MINISTERUL EDUCAȚIEI ȘI ÎNVĂȚĂMÎNTULUI

IX

MATEMATICĂ APLICATĂ ÎN TEHNICA DE CALCUL

clasa a IX-a

Manual pentru licee cu profil
de matematică și de matematică-fizică

1011
1101
0011
1101
1101

$\pi = 3{,}1415926\ldots$

PVTP

$\pi^{*} = 3{,}14$

EDITURA DIDACTICĂ ȘI PEDAGOGICĂ
BUCUREȘTI, 1987

MINISTERUL EDUCAȚIEI ȘI ÎNVĂȚĂMÎNTULUI

IOANA BĂRBAT

Dr. ALEXANDRU DUMITRACHE

MATEMATICĂ APLICATĂ ÎN TEHNICA DE CALCUL

clasa a IX-a

Manual pentru licee cu profil
de matematică și de matematică-fizică

EDITURA DIDACTICĂ ȘI PEDAGOGICĂ
BUCUREȘTI

Manualul a fost elaborat în 1980 și revizuit în anul 1982 pe baza programei școlare aprobate de Ministerul Educației și Învățământului.

Coordonator : *Lector univ. dr.* IOAN TOMESCU

Referenți : *Prof. univ. dr.* C. P. POPOVICI

Catedra de matematică a liceului Matei Basarab din București

Prof. IOAN MAFTEI

Redactor : *Prof.* ELEONORA DRĂGHIA

Tehnoredactor : VICTORIA GHIMIȘ

CUPRINS

Capitolul I

I. SISTEME DE NUMERAȚIE 5
I.1. Definirea sistemelor de numerație 5
I.2. Scrierea unui număr real într-un sistem de numerație 6
I.3. Sistemul de numerație în baza zece 8
I.4. Sistemul de numerație în baza doi 9
I.5. Sistemul de numerație în baza opt 11
I.6. Sistemul de numerație în baza șaisprezece . . . 12
I.7. Trecerea numerelor dintr-o bază în alta 15
Exerciții și probleme 25

Capitolul II

II. PRELUCRAREA AUTOMATĂ A DATELOR . . . 28
II.1. Informația și informatica 28
II.2. Codificarea informațiilor 29
II.3. Arhitectura și structura unui sistem de calcul . 30
II.4. Limbaje de programare 32
II.5. Dezvoltare de programe 33
II.6. Informatica în România 34

Capitolul III

III. OPERAȚII ARITMETICE ȘI OPERAȚII LOGICE 37
III.1. Noțiuni introductive 37
III.2. Aproximantele numerelor reale 37
III.3. Procedee de aproximare a numerelor reale . 38
III.4. Erori și tipuri de erori 41
III.5. Erori datorate modului de reprezentare . . 44
III.6. Inegalitățile fundamentale în aproximarea numerelor reale 45
III.7. Aproximante cu eroarea absolută mai mică sau egală cu $\frac{1}{10^k}$ 47
III.8. Cifre sigure și cifre îndoielnice pentru o aproximantă dată 49
III.9. Determinarea erorilor relative maxime ale unei aproximante pe baza cifrelor sigure ale acesteia 51

III.10. Determinarea numărului de cifre sigure ale unei aproximante cînd se cunoaşte eroarea relativă maximă a ei 53
III.11. Efectuarea sumelor şi diferenţelor cu aproximante 55
III.12. Efectuarea produselor şi cîturilor cu aproximante 61
III.13. Ridicarea la putere a aproximantelor . . . 65
Exerciţii şi probleme 67
III.14. Operaţii logice 73
III.14.1. Introducere 73
III.14.2. Calculul propoziţiilor 75
III.14.2.1. Diagrame Euler-Venn 78
III.14.2.2. Tautologii şi contradicţii . . . 79
III.14.2.3. Raţionament 83
III.14.2.4. Tipuri de raţionamente . . . 84
Întrebări şi exerciţii 88
III.14.3. Calculul predicatelor 90
III.14.3.1. Calculul predicatelor ca dezvoltare a calculului propoziţiilor . 90
III.14.3.2. Particularizarea unui predicat. Cuantificări 93
III.14.3.3. Operaţii cu predicate 96
III.14.3.4. Identităţi logice 98
Întrebări şi exerciţii 103
Bibliografie 106

I
SISTEME DE NUMERAŢIE

I.1. Definirea sistemelor de numeraţie

Pentru reprezentarea numerelor se aleg două mulţimi, una A de denumiri specifică fiecărei limbi şi a doua B de simboluri distincte, numite cifre, între care să existe o corespondenţă biunivocă.

I.1.1. *Definiţie.* **Numărul $b > 1$ al cifrelor mulţimii B se numeşte baza sistemului de numeraţie. Numerele naturale c pentru care $0 \leqslant c \leqslant b - 1$ se numesc cifre în baza b.**

Pentru scrierea numerelor în baza b se folosesc b semne distincte.

Fie N numărul de elemente ale unei mulţimi M. Vom grupa elementele lui M în submulţimi, avînd fiecare cîte b elemente, obţinînd k_1 de astfel de submulţimi şi încă o submulţime cu a_0 elemente cu $0 \leqslant a_0 < b$.

Putem scrie:

$$N = k_1 b + a_0. \tag{1}$$

Pentru cazul $N < b$, avem $k_1 = 0$ şi deci nu se poate forma nici o submulţime cu b elemente. Avem $N = a_0$.

În cazul $k_1 < b$ păstrăm N sub forma $N = k_1 b + a_0$.

Pentru $k_1 \geqslant b$ procedăm cu k_1 întocmai cum am procedat anterior cu N găsind k_2 grupe de cîte b submulţimi cu cîte b elemente şi a_1 submulţimi de cîte b elemente, în care $0 \leqslant a_1 < b$.

Putem scrie pentru k_1 o relaţie analoagă cu (1), scrisă pentru N,

$$k_1 = k_2 b + a_1. \tag{2}$$

Dar ţinînd seama de (1) şi (2), avem

$$N = (k_2 b + a_1)b + a_0\,;\quad N = k_2 b^2 + a_1 b + a_0.$$

Dacă avem $k_2 \geqslant b$ continuăm procesul formării de grupe de submulţimi obţinînd k_3 grupe de cîte b grupe de cîte b submulţimi cu cîte b elemente, şi încă a_2 grupe de cîte b grupe de submulţimi cu b elemente.

Se obţine astfel:

$$N = [(k_3 b + a_2)b + a_1]b + a_0$$

adică,

$$N = k_3 b^3 + a_2 b^2 + a_1 b + a_0.$$

Continuînd, prin același procedeu, deoarece mulțimea M are N elemente, se ajunge după un număr finit de pași la un $k_n = a_n$, unde $1 \leqslant a_n \leqslant b - 1$ și deci

$$N = a_n b^n + a_{n-1} b^{n-1} + \ldots + a_2 b^2 + a_1 b + a_0. \tag{3}$$

Un număr scris sub forma (3) se spune că este scris sub *formă polinomială*.

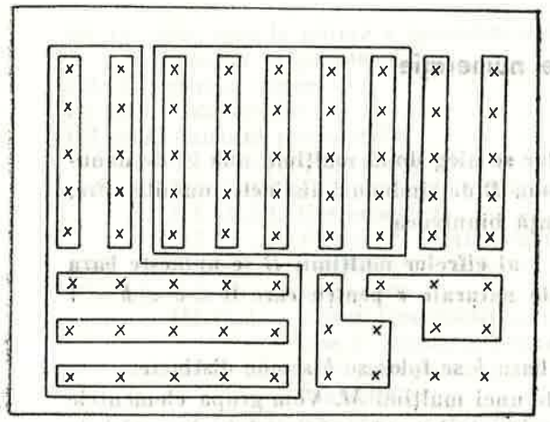


Fig. I.1

Exemplu. În figura I.1 mulțimea N este mulțimea tuturor punctelor încadrate iar $b = 5$. Se observă că $N = 2 \cdot 5^2 + 4 \cdot 5 + 2$.

Dacă baza b este precizată și nu există posibilitatea unor confuzii, atunci numărul N poate fi scris într-o formă mai simplă numai cu ajutorul simbolurilor $a_n, a_{n-1}, \ldots, a_1, a_0$ care formează fiecare o singură cifră și $a_n \neq$ astfel :

$$N = a_n a_{n-1} a_{n-2} \ldots a_1 a_0. \tag{4}$$

Fiecare dintre cifrele $a_n, a_{n-1}, \ldots, a_1, a_0$ ocupă o anumită poziție indicată de exponentul bazei. Scrierea sub forma (4) mai poartă denumirea de *scriere pozițională*.

Introducem noțiunea de *ordin* al unei cifre care este indicat de exponentul bazei din produsul dintre cifră și o putere a bazei. În (4) cifra a_0 este cifra de ordinul zero, cifra a_1 este cifra de ordinul 1 și așa mai departe, cifra a_n este de ordinul n. În acest fel citirea numărului N se poate face și cifră cu cifră începînd de la stînga spre dreapta, adică de la ordinul cel mai mare n pînă la ordinul cel mai mic zero.

I.2. Scrierea unui număr real într-un sistem de numerație

În paragraful precedent s-a arătat cum se scrie un număr natural, într-o bază de numerație b oarecare. Se știe însă că orice număr întreg este de forma $\pm N$ unde N este număr natural. Deci pentru un întreg M oarecare, scrierea într-o bază dată b este :

$$M = \pm(a_n b^n + a_{n-1} b^{n-1} + \ldots + a_1 b + a_0) \tag{5}$$

sau

$$M = \pm a_n a_{n-1} \ldots a_1 a_{0(b)}. \tag{6}$$

Un număr real oarecare este suma dintre un număr întreg și un număr fracționar pozitiv, numite partea întreagă și partea fracționară, scriindu-se în ordine : partea întreagă, virgula și apoi partea fracționară*.

Exemple :

$$3 + 0{,}21 = 3{,}21 ;$$

$$-4 + 0{,}62 = \overline{4}{,}62.$$

Pentru numerele reale negative se utilizează frecvent scrierea în care semnul minus (—) scris la stînga numărului este atribuit și părții întregi și părții fracționare.

Exemplu :

$$\overline{5}{,}24 = -5 + 0{,}24 = (-4) + (-1 + 0{,}24) = (-4) - (0{,}76) = -4{,}76.$$

De aceea, pentru claritatea înțelegerii, în cele ce urmează, vom numi întregii unui număr, partea de la stînga virgulei și parte fracționară (sau parte zecimală) partea de la dreapta virgulei, în toate cazurile în care avem același semn și pentru întregi și pentru partea fracționară.

Pentru un număr real oarecare x, care are și întregi și parte fracționară, vom avea o scriere de forma

$$x = \pm\left(a_n b^n + a_{n-1} b^{n-1} + \ldots + a_1 b + a_0 + \frac{a_{-1}}{b} + \frac{a_{-2}}{b^2} + \ldots + \frac{a_{-m}}{b^m}\right) + r_m, \tag{7}$$

unde

$$0 \leqslant a_i < b \text{ pentru orice } i = 0, 1, \ldots, n, \text{ cu } a_n \neq 0$$

și

$$0 \leqslant a_{-j} < b \text{ pentru orice } j = 0, 1, \ldots, m,$$

iar restul r_m verifică inegalitățile

$$0 \leqslant r_m < \frac{1}{b^m}.$$

Este necesar să introducem un semn de separare a întregilor de partea fracționară. În mod obișnuit se folosește virgula, numită și virgula zecimală, însă trebuie știut că se folosește și punctul, numit punct zecimal (în scrierea anglo-saxonă).

Deci un număr x cu $r_m = 0$ se scrie

$$x = \pm a_n a_{n-1} \ldots a_1 a_0, \; a_{-1} a_{-2} \ldots a_{-m(b)} \tag{8}$$

sau

$$x = \pm a_n a_{n-1} \ldots a_1 a_0 . a_{-1} a_{-2} \ldots a_{-m(b)}.$$

* Algebra cl. a IX-a, cap. III.

Exemple :

1) Numărul $257_{(b)}$, $(b > 7)$, (se citește cifră cu cifră), se scrie

$$257_{(b)} = 2b^2 + 5b + 7.$$

2) Numărul $-586_{(b)}$, $(b > 8)$, (se citește cifră cu cifră), se scrie

$$-586_{(b)} = -(5 \cdot b^2 + 8b + 6).$$

3) Numărul $x = 3b^2 + 7b + 4 + \frac{4}{b} + \frac{5}{b^2} + \frac{2}{b^4}$ scris în baza b $(b > 7)$ este $x = 374{,}4502_{(b)}$.

I.3. Sistemul de numerație în baza zece

Sistemul de numerație în baza zece, numit și sistem zecimal, folosește zece cifre distincte date prin simbolurile 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, cu ajutorul cărora putem scrie orice număr.

Orice număr natural mai mic decît zece se reprezintă prin una din aceste cifre și arată cîte unități de ordinul zero are numărul.

Un număr natural N, mai mare sau egal cu zece, se va reprezenta printr-un șir de cifre în felul următor :

Numărul natural zece, egal deci cu numărul cifrelor bazei utilizate, se reprezintă prin 10 și se citește zece. Acest număr se numește unitate de ordinul întîi și reprezintă zece unități de ordinul zero.

Un număr care conține a_1 unități de ordinul întîi și a_0 unități de ordinul zero se scrie

$$a_1a_0 = a_1 \cdot 10 + a_0.$$

Un număr care conține 10 unități de ordinul întîi se scrie 100 și reprezintă o unitate de ordinul doi. Unitatea de ordinul doi este produsul între baza zece și unitatea de ordinul întîi

$$10 \cdot 10 = 10^2 = 100.$$

Un număr care conține a_2 unități de ordinul doi, a_1 unități de ordinul întîi și a_0 unități de ordinul zero se scrie astfel :

$$a_2a_1a_0 = a_2 \cdot 10^2 + a_1 \cdot 10 + a_0.$$

Generalizînd, putem spune că un număr care conține 10 unități de ordinul k, se va numi unitate de ordinul $k + 1$ și se va scrie astfel :

$$10 \cdot 10^k = 10^{k+1} = \underbrace{10\ldots0}_{k+1 \text{ cifre de zero}}.$$

Numărul ce conține a_n unități de ordinul n, a_{n-1} unități de ordinul $n-1$ și așa mai departe, a_1 unități de ordinul întîi și a_0 unități de ordinul zero se scrie,

$$a_na_{n-1}\ldots a_1a_0 = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \ldots + a_1 \cdot 10 + a_0,$$

unde

$$0 \leqslant a_i \leqslant 9 \text{ pentru } i = 0, 1, \ldots, n, \text{ cu } a_n \neq 0.$$

Ca și în cazul bazei b putem extinde scrierea în baza zece la numere reale.

Pentru un număr întreg M care este de forma $\pm N$, unde N este număr natural, avem scrierea :

$$M = \pm(a_n10^n + a_{n-1}10^{n-1} + \ldots + a_110 + a_0)$$

sau

$$M = \pm a_na_{n-1}\ldots a_1a_0,$$

iar pentru un număr real x oarecare, avînd și parte zecimală, avem :

$$x = \pm\left(a_n10^n + a_{n-1}10^{n-1} + \ldots + a_110 + a_0 + \frac{a_{-1}}{10} + \frac{a_{-2}}{10^2} + \ldots \right.$$
$$\left. \ldots + \frac{a_{-m}}{10^m} + \ldots\right),$$

unde

$$0 \leqslant a_i \leqslant 9 \text{ pentru } i = 0, 1, \ldots, n, \text{ cu } a_n \neq 0$$

și

$$0 \leqslant a_{-j} \leqslant 9 \text{ pentru } j = 0, 1, \ldots, m, \ldots$$

sau

$$x = \pm\, a_na_{n-1}\ldots a_1a_0,a_{-1}a_{-2}\ldots a_{-m}\ldots$$

unde virgula separă întregii de partea zecimală.

La scrierea în baza zece de obicei nu se mai indică alăturat baza așa cum am procedat și în cazurile de mai sus.

I.4. Sistemul de numerație în baza doi

I.4.1. Scrierea numerelor în baza doi

Sistemul de numerație în baza doi, numit și sistem de numerație binar, este sistemul în care se folosesc cifrele 0 și 1.

Numerele naturale mai mici decît doi se vor scrie cu aceste cifre, respectiv „0“ pentru zero și „1“ pentru unu.

Baza de numerație fiind doi o vom nota $10_{(2)}$ și pentru un număr natural oarecare avem:

$$N = a_n 10^n_{(2)} + a_{n-1} 10^{n-1}_{(2)} + \ldots + a_1 10_{(2)} + a_0,$$

unde

$$0 \leqslant a_i \leqslant 1 \text{ pentru } i = 0, 1, \ldots, n-1 \text{ și } a_n = 1$$

sau

$$N = a_n a_{n-1} \ldots a_1 a_{0(2)}$$

cu aceleași condiții pentru a_i și a_n.

Un număr care conține o unitate de ordinul zero se scrie $N = 1$.

Un număr care conține o unitate de ordinul întîi se scrie $N = 10_{(2)}$.

Un număr care conține o unitate de ordinul întîi și o unitate de ordinul zero se scrie

$$N = 11_{(2)}.$$

În general, o unitate de ordinul n conține două unități de ordinul $n-1$, adică $10_{(2)} \cdot 10^{n-1}_{(2)} = 10^n_{(2)}$ sau $2 \cdot 2^{n-1} = 2^n$ în baza zece.

Folosind rezultatele date de formulele (7) și (8) din paragraful I.2 pentru cazul $b = 2$, scriem un număr real x în baza doi sub forma:

$$x = \pm \left(a_n 10^n_{(2)} + a_{n-1} 10^{n-1}_{(2)} + \ldots + a_1 10_{(2)} + a_0 + \frac{a_{-1}}{10_{(2)}} + \frac{a_{-2}}{10^2_{(2)}} + \ldots \right.$$
$$\left. \ldots + \frac{a_{-m}}{10^m_{(2)}} + \ldots \right),$$

unde

$$0 \leqslant a_i \leqslant 1 \text{ pentru } i = 0, 1, \ldots, n-1, \; a_n = 1$$

și

$$0 \leqslant a_{-j} \leqslant 1 \text{ pentru } j = 0, 1, \ldots, m, \ldots$$

sau

$$x = \pm\, a_n a_{n-1} \ldots a_1 a_0, a_{-1} a_{-2} \ldots a_{-m} \ldots_{(2)}.$$

I.4.2. Operații cu numere scrise în baza doi

În general efectuarea operațiilor în baza doi nu se deosebește esențial de efectuarea operațiilor în baza zece. Se păstrează formele de așezare a numerelor pentru efectuarea operațiilor și numai tabelele de operații sînt altele. Deoarece urmărirea operațiilor cu numere scrise în binar devine mult mai ușoară dacă se cunosc tabelele de operații pentru adunare și înmulțire, le dăm pe acestea în continuare.

Tabelul I.1

+	0	1
0	0	1
1	1	10

Tabelul I.2

×	0	1
0	0	0
1	0	1

Să urmărim în continuare cîteva exemple:

1)
```
  11,111 +
  10,001
  ------
 110,000
```

2)
```
 100000,011 −
   1000,101
 ---------
  10111,110
```

3)
```
   11,01 ×
    1,01 1
   ------
    11 01
   11 0 1
  11 01
  ---------
 100,01111
```

4)
```
 1101,1 | 110
 110    |-----
 ---------| 10,01
   1 10
   1 10
   ---
```

I.5. Sistemul de numerație în baza opt

I.5.1. Scrierea numerelor în baza opt

Sistemul de numerație în baza opt, numit și sistem de numerație octal, este sistemul în care se folosesc opt cifre și anume, 0, 1, 2, 3, 4, 5, 6, 7 iar baza se notează cu ajutorul cifrelor „0" și „1" scriind 10, dacă nu este posibilă o confuzie, în caz contrar scriind cu specificație $10_{(8)}$.

Avem:

$$N = a_n 10^n_{(8)} + a_{n-1} 10^{n-1}_{(8)} + \ldots + a_1 10_{(8)} + a_0$$

unde

$$0 \leqslant a_i \leqslant 7 \text{ pentru } i = 0, 1, \ldots, n \text{ cu } a_n \neq 0$$

sau

$$N = a_n a_{n-1} \ldots a_1 a_{0(8)}$$

cu aceleași condiții pentru a_i și a_n.

Numerele mai mici decît opt se vor scrie cu cifrele indicate pentru această bază și vor fi numite unități de ordinul zero.

O unitate de un anumit ordin (în afară de ordinul zero) este egală cu opt unități de ordin inferior iar opt unități de un anumit ordin formează o unitate de ordin superior, adică,

$$10_{(8)} \cdot 10^{n-1}_{(8)} = 10^n_{(8)} \text{ sau în baza zece } 8 \cdot 8^{n-1} = 8^n.$$

Formulele (7) și (8) din paragraful I.2 pentru cazul $b = 8$ ne permit să scriem un număr real x în baza opt sub forma:

$$x = \pm \left(a_n 10^n_{(8)} + a_{n-1} 10^{n-1}_{(8)} + \ldots + a_1 10_{(8)} + a_0 + \frac{a_{-1}}{10_{(8)}} + \frac{a_{-2}}{10^2_{(8)}} + \ldots \right.$$
$$\left. \ldots + \frac{a_{-m}}{10^m_{(8)}} + \ldots \right),$$

unde

$$0 \leqslant a_i \leqslant 7 \text{ pentru } i = 0, 1, \ldots, n \text{ cu } a_n \neq 0$$

și

$$0 \leqslant a_{-j} \leqslant 7 \text{ pentru } j = 0, 1, \ldots, m, \ldots$$

sau

$$x = a_n a_{n-1} \ldots a_1 a_0, a_{-1} a_{-2} \ldots a_{-m} \ldots_{(8)}.$$

I.5.2. Operații cu numere scrise în baza opt

Și în baza opt, ca și în baza doi, efectuarea operațiilor nu se deosebește prea mult de efectuarea operațiilor în baza zece, deoarece se păstrează formele de așezare ale numerelor. Diferă numai tabelele de operații și de aceea pentru a da posibilitatea unei urmăriri mai ușoare le dăm în continuare :

Tabelul I.3

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Tabelul I.4

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Să urmărim mai jos cîteva exemple de operații în octal.

1)
$$\begin{array}{r} 547{,}317 + \\ 3250{,}715 \\ \hline 4020{,}234 \end{array}$$

2)
$$\begin{array}{r} 7456{,}625 - \\ 6767{,}546 \\ \hline 467{,}057 \end{array}$$

3)
$$\begin{array}{r} 26{,}5 \times \\ 32{,}4 \\ \hline 1\,3\,2\,4 \\ 5\,5\,2 \\ 1\,0\,3\,7 \\ \hline 1\,1\,2\,7{,}4\,4 \end{array}$$

b)
$$\begin{array}{r|l} 30{,}6 & 26 \\ \hline 26 & 1{,}1 \\ \hline =26 & \\ 26 & \\ \hline == & \end{array}$$

I.6. Sistemul de numerație în baza șaisprezece

I.6.1. Scrierea numerelor în baza șaisprezece

Baza de numerație șaisprezece, numită și hexazecimală, folosește șaisprezece semne pentru cifrele bazei. Primele zece semne sînt cele cunoscute de la baza zece, iar în continuare sînt utilizate literele A, B, C, D, E, F, care au următoarele semnificații :

— cifra A în sistemul hexazecimal reprezintă numărul 10 din sistemul zecimal ;

— cifra B în sistemul hexazecimal reprezintă numărul 11 din sistemul zecimal ;

— cifra C în sistemul hexazecimal reprezintă numărul 12 din sistemul zecimal ;

— cifra D în sistemul hexazecimal reprezintă numărul 13 din sistemul zecimal ;

— cifra E în sistemul hexazecimal reprezintă numărul 14 din sistemul zecimal ;

— cifra F în sistemul hexazecimal reprezintă numărul 15 din sistemul zecimal.

Baza se notează cu ajutorul cifrelor „0" și „1" scriind 10, dacă nu este posibilă vreo confuzie ; în caz contrar baza se scrie cu specificație $10_{(16)}$.

Numerele naturale mai mici decît baza se vor scrie cu una din cifrele indicate.

Pentru un număr natural oarecare avem :

$$N = a_n 10^n_{(16)} + a_{n-1} 10^{n-1}_{(16)} + \ldots + a_1 10_{(16)} + a_0,$$

unde

$$0 \leqslant a_i \leqslant F \text{ pentru } i = 0, 1, \ldots, n \text{ cu } a_n \neq 0,$$

sau

$$N = a_n a_{n-1} \cdots a_1 a_{0(16)}$$

cu aceleași condiții pentru a_i și a_n.

Orice unitate de un anumit ordin este egală cu șaisprezece unități de ordin inferior (cu excepția ordinului zero) iar șaisprezece unități de un anumit ordin formează o unitate de ordin superior, adică

$$10_{(16)} \cdot 10^{n-1}_{(16)} = 10^n_{(16)} \text{ sau în baza zece } 16 \cdot 16^{n-1} = 16^n.$$

Ca și în cazul celorlalte baze de numerație de pînă acum, putem aplica formulele (7) și (8) din paragraful I.2 pentru cazul $b = 16$ și putem scrie un număr real x în baza șaisprezece sub forma,

$$x = \pm \left(a_n 10^n_{(16)} + a_{n-1} 10^{n-1}_{(16)} + \ldots + a_1 10_{(16)} + a_0 + \frac{a_{-1}}{10_{(16)}} + \frac{a_{-2}}{10^2_{(16)}} + \cdots \right.$$
$$\left. \ldots + \frac{a_{-m}}{10^m_{(16)}} + \ldots \right),$$

unde

$$0 \leqslant a_i \leqslant F \text{ pentru } i = 0, 1, \ldots, n \text{ cu } a_n \neq 0$$

și

$$0 \leqslant a_{-j} \leqslant F \text{ pentru } j = 0, 1, \ldots, m, \ldots$$

sau

$$x = \pm a_n a_{n-1} \ldots a_1 a_0, a_{-1} a_{-2} \ldots a_{-m} \ldots_{(16)}.$$

I.6.2. Operații cu numere scrise în baza șaisprezece

Și în hexazecimal operațiile se fac ca în bazele doi, opt sau zece. De aceea vom da mai întîi tabelele de operații pentru adunare și înmulțire.

Tabelul I.5

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Tabelul I.6

×	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Exemple de operații în hexazecimal.

1) FFC,ED + 14A,2B = 1 147,18

2) A02,B4 − 87B,FC = 186,B8

3) A,7 × 8,C = 7D4 + 538 = 5B,54

4) C8 : A = 14 (A × 14 = 28, 28 = 28, 0)

Sistemele de numerație binar, octal și hexazecimal precum și operațiile cu numere scrise în aceste sisteme sînt utilizate în toate problemele ce sînt rezolvate cu ajutorul calculatoarelor. Așa cum se va vedea în capitolele următoare, scrierea în sistemul binar asigură posibilitatea de introducere în calculator a datelor și comenzilor pentru faptul că utilizînd doar două simboluri (0 și 1), acestea pot fi ușor modelate fizic.

I.7. Trecerea numerelor dintr-o bază în alta

I.7.1. Trecerea unui număr dintr-o bază în alta prin baza zece

Trecerea numerelor dintr-o bază de numerație în alta se mai numește și *conversie*. La tratarea acestei probleme putem presupune, pentru simplificare, că numărul este întreg sau chiar natural.

Fie N scris în baza b și vrem să-l scriem în baza β diferită de b. În general, trecerea de la o bază la alta se face prin intermediul bazei zece. De aceea vom scrie atît cifrele numărului N cît și baza b în baza zece.

Dacă numărul este :

(9) $N = a_n b^n + a_{n-1} b^{n-1} + \ldots + a_1 b + a_0$, cu $a_n \neq 0$ și $i = 0, 1, \ldots, n$,

unde a_i scrise în baza zece sînt de forma

$$a_i = a'_{j_i} a'_{j_i - 1} \ldots a'_1 a'_0 \text{ cu } 0 \leqslant a'_l \leqslant 9 \text{ și } l = 0, 1, \ldots, j_i,$$

iar baza b scrisă în bază zece are forma

$$b = a''_k a''_{k-1} \ldots a''_1 a''_0 \text{ cu } 0 \leqslant a''_l \leqslant 9 \text{ și } a''_k \neq 0,$$

atunci înlocuind în (8) și efectuînd operațiile în baza zece obținem numărul sub forma

$$N = \bar{a}_m 10^m + \bar{a}_{m-1} 10^{m-1} + \ldots + \bar{a}_1 10 + \bar{a}_0,$$

sau

$$N = \bar{a}_m \bar{a}_{m-1} \ldots \bar{a}_1 \bar{a}_0.$$

Pentru exemplificare, să luăm

$$N = CD82_{(16)},$$

adică

$$N = C\cdot 10^3_{(16)} + D\cdot 10^2_{(16)} + 8\cdot 10_{(16)} + 2.$$

Scriindu-i cifrele și baza în baza zece, avem

$$C = 12\,;\ D = 13\,;\ 8 = 8\,;\ 2 = 2\,;\ 10_{(16)} = 16.$$

Deci

$$N = 12\cdot 16^3 + 13\cdot 16^2 + 8\cdot 16 + 2$$

unde făcînd calculele obținem :

$$N = 5\cdot 10^4 + 2\cdot 10^3 + 6\cdot 10^2 + 1\cdot 10 = 52\,610.$$

Pentru a trece numărul

$$N = \bar{a}_m\bar{a}_{m-1}\ldots\bar{a}_1\bar{a}_0$$

din baza zece în baza β trecem la numărarea unei mulțimi cu N elemente, N fiind scris în baza zece, formînd submulțimi de cîte β elemente și grupe cu β submulțimi de cîte β elemente etc.

Realizăm mai ușor această numărare scriind și baza β în baza zece și apoi efectuăm împărțirea lui N scris în baza zece la β scris în baza zece și apoi cîtul obținut îl împărțim din nou la β scris în baza zece și așa mai departe.

Această operație se desfășoară după următorul algoritm :

$$N = k_1\beta + \alpha_0, \qquad 0 \leqslant \alpha_0 < \beta,$$

$$k_1 = k_2\beta + \alpha_1, \qquad 0 \leqslant \alpha_1 < \beta,$$

$$k_2 = k_3\beta + \alpha_2, \qquad 0 \leqslant \alpha_2 < \beta,$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$k_{p-1} = k_p\beta + \alpha_{p-1}, \qquad 0 \leqslant \alpha_{p-1} < \beta,$$

$$k_p = 0\cdot\beta + \alpha_p, \qquad \alpha_p = k_p,$$

ultima împărțire efectuată fiind aceea în care $k_p < \beta$. Numărul N scris în baza β va fi deci

$$N = \alpha_p\alpha_{p-1}\ldots\alpha_1\alpha_0.$$

Exemplu :

Fie $N = 52\,610$ în baza zece pe care să-l scriem în baza 8.
Putem utiliza o schemă de împărțire puțin modificată (fig. I.2) astfel

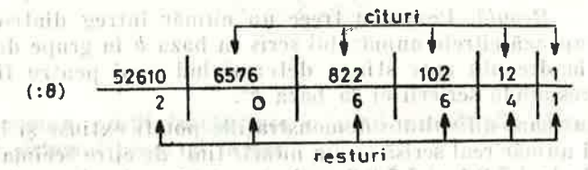


Fig. I.2

Deci $52\,610 = 146\,602_{(8)}$.

I.7.2. Trecerea unui număr dintr-o bază b^k în baza b și invers

În trecerea de la baza b la baza β, un interes deosebit îl prezintă cazul în care între b și β există una dintre relațiile :

$$b = \beta^k \text{ sau } \beta = b^k.$$

Să presupunem că avem $\beta = b^k$.

În acest caz, numărul N dat de expresia

$$N = \alpha_m\beta^m + \alpha_{m-1}\beta^{m-1} + \ldots + a_1\beta + \alpha_0 \tag{10}$$

poate fi scris în baza b punînd în locul bazei β pe b^k, iar cifrele α_i, ($i = 0, 1, \ldots, m$) din baza β vor fi scrise în baza b.

Cifrele α_i scrise în baza b sînt numere cu k cifre de forma :

$$\alpha_i = \alpha_{i,k-1}b^{k-1} + \alpha_{i,k-2}b^{k-2} + \ldots + \alpha_{i,1}b + \alpha_{i,0},$$

unde coeficienții $\alpha_{i,h}$, ($i = 0, 1, \ldots, m$; $h = 0, 1, \ldots, k-1$) sînt toți nuli sau nu după cum α_i au fost nuli sau nu, deoarece $0 \leqslant \alpha_i \leqslant \beta - 1$.

Înlocuind exprimările lui α_i în (10) se obține :

$$N = (\alpha_{m,k-1}b^{k-1} + \alpha_{m,k-2}b^{k-2} + \ldots + \alpha_{m,1}b + \alpha_{m,0})b^{mk} +$$

$$+ (\alpha_{m-1,k-1}b^{k-1} + \ldots + \alpha_{m-1,0})b^{(m-1)k} + \ldots + (\alpha_{1,k-1}b^{k-1} + \alpha_{1,k-2}b^{k-2} +$$

$$+ \ldots + \alpha_{1,1}b + \alpha_{1,0})b^k + (\alpha_{0,k-1}b^{k-1} + \alpha_{0,k-2}b^{k-2} + \ldots + \alpha_{0,1}b + \alpha_{0,0}),$$

adică

$$N = \alpha_{m,k-1}b^{mk+k-1} + \alpha_{m,k-2}b^{mk+k-2} + \ldots + \alpha_{0,1}b + \alpha_{0,0}.$$

De unde rezultă că cifrele lui N în baza b sînt cele găsite pentru fiecare α_m din scrierea în baza β fără modificarea ordinei.

Am obținut deci următoarea

I.7.2.1. — *Regulă.* Pentru trecerea unui număr întreg N de la o bază b^k la baza b scriem fiecare dintre cifrele numărului N din baza b^k în baza b cu

ajutorul a exact k cifre în baza b, păstrînd ordinea cifrelor din scrierea în baza b^k și avînd grijă ca prima cifră din stînga a numărului obținut să fie nenulă.

Nu este greu de observat că și trecerea inversă se face după următoarea

I.7.2.2. — *Regulă.* Pentru a trece un număr întreg dintr-o bază b în baza b^k se grupează cifrele numărului scris în baza b în grupe de cîte k cifre începînd de la dreapta spre stînga determinînd apoi pentru fiecare grupă cifra care corespunde scrierii ei în baza b^k.

Fără prea mare dificultate demonstrațiile pot fi extinse și la partea zecimală a unui număr real scrisă cu un număr finit de cifre zecimale și în acest caz regulile de la I.7.2.1 și I.7.2.2 aplicate părții zecimale cu începere de la virgulă spre dreapta ne produc trecerea unui număr real dintr-o bază b^k în baza b și invers.

I.7.3. Trecerea numerelor întregi din sistemul zecimal în sistemul binar și invers

Reamintindu-ne rezultatul obținut în paragraful I.7.1 putem repede reconstitui algoritmul de trecere de la baza 10 la baza 2 și anume :

I.7.3.1. — *Regulă.* Pentru trecerea unui număr întreg din baza 10 în baza 2 se împarte numărul scris în baza 10 la 2 găsindu-se restul 0 sau 1, apoi cîtul obținut se împarte iar la 2 găsindu-se restul 0 sau 1 și așa mai departe, pînă cînd ultimul cît obținut va fi 0, deci ultimul rest este egal cu 1. Cifrele care reprezintă resturile, luate în ordinea inversă obținerii, formează numărul căutat.

Exemple :

Să se facă conversia din baza 10 în baza 2 a numerelor :

a) 75 ; b) −379.

Pentru calcul utilizăm o așezare ca în figura I.3.

a) (2) | 75 | 37 | 18 | 9 | 4 | 2 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | ; $75 = 1001011_{(2)}$

b) (2) | 379 | 189 | 94 | 47 | 23 | 11 | 5 | 2 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | ; $-379 = -101111011_{(2)}$

Fig. I.3

Pentru conversia din baza 2 în baza 10 folosim scrierea :

$$N = a_n 10_{(2)}^n + a_{n-1} 10_{(2)}^{n-1} + \ldots + a_1 10_{(2)} + a_0.$$

Știind că $10_{(2)} = 2$ și că oricare a_i este cifra 0 sau 1 aceeași și în baza zece deducem :

$$N = \sum_{i=0}^{n} a_i 2^i.$$

I.7.4. Trecerea părții fracționare a unui număr din sistemul zecimal în sistemul binar și invers

Fie x numărul ce reprezintă o parte fracționară în sistemul zecimal pe care vrem să o trecem în binar. Scrierea în binar a lui x, avînd $10_{(2)}$ exprimat în baza zece, deci $10_{(2)} = 2$, este :

$$x = \frac{a_{-1}}{2} + \frac{a_{-2}}{2^2} + \ldots + \frac{a_{-m}}{2^m}.$$

Dacă înmulțim pe x cu 2 obținem :

$$2x = a_{-1} + \frac{a_{-2}}{2} + \ldots + \frac{a_{-m}}{2^{m-1}}.$$

Se observă că prima cifră de la partea fracționară a_{-1} devine cifră de la partea întreagă a numărului înmulțit cu 2. Este ușor acum de continuat. După înmulțiri succesive cu 2 putem pune în evidență pe rînd cifrele de la partea fracționară a numărului scris în baza 2. Avem așadar :

I.7.4.1. — *Regulă.* Pentru determinarea cifrelor din scrierea unei părți fracționare la trecerea din baza zece în baza doi, se înmulțește numărul cu 2 și partea întreagă a numărului obținut va fi prima cifră de după virgulă a numărului scris în binar, apoi se înmulțește cu 2 partea fracționară a produsului obținut anterior și partea întreagă a noului rezultat va fi a doua cifră de după virgulă a numărului scris în baza 2 ș.a.m.d.

Menționăm că și în baza 2 pot fi părți fracționare periodice.

Exemplu :

Să se facă conversia din zecimal în binar a numărului 0,375.

Utilizăm schema din figura I.4.

a) (·2) | 375 | 750 | 50 | 0 |
| 0 | 0 | 1 | 1 | ; $0,375 = 0,011_{(2)}$

partea întreagă

Fig. I.4

Pentru trecerea de la baza 2 la baza 10 considerăm scrierea numărului x, parte fracționară, în baza 2

$$x = \frac{a_{-1}}{10_{(2)}} + \frac{a_{-2}}{10^2_{(2)}} + \ldots + \frac{a_{-m}}{10^m_{(2)}}$$

în care trecem pe $10_{(2)}$ în baza zece și avem $10_{(2)} = 2$, iar cifrele numărului x din baza 2 au aceeași semnificație și în baza zece. Urmează că :

$$x = \frac{a_{-1}}{2} + \frac{a_{-2}}{2^2} + \ldots + \frac{a_{-m}}{2^m},$$

adică

$$x = \sum_{i=1}^{m} \frac{a_{-i}}{2^i}.$$

Exemplu :

Să se facă conversia din binar în zecimal pentru numărul $0{,}01011_{(2)}$.

Avem :

$$0{,}01011_{(2)} = \frac{1}{2^2} + \frac{1}{2^4} + \frac{1}{2^5} = \frac{11}{32} = 0{,}34375.$$

Observație : Dacă un număr are și întregi și parte fracționară, atunci conversia se face separat pentru fiecare parte, apoi se reconstituie numărul din părțile transformate.

Exemplu :

$$42{,}13 = 42 + 0{,}13 = 101010_{(2)} + 0{,}00(10000101000111101011)_{(2)} =$$
$$= 101010{,}00(10000101000111101011)_{(2)},$$

unde grupul de cifre în paranteze de la partea fracționară reprezintă perioada acestui număr în baza 2.

I.7.5. Trecerea numerelor întregi din sistemul zecimal în sistemul octal sau hexazecimal și invers

Bazîndu-ne pe același rezultat obținut în paragraful I.7.1 putem reconstitui algoritmul de trecere de la baza 10 la baza 8 sau 16 ca și în cazul bazei 2. Putem prelua regula I.7.3.1 în care în locul bazei 2 avem în vedere baza 8 sau 16 în rest totul rămîne neschimbat.

Exemplul tratat la I.7.1 folosind figura I.2 arată trecerea de la baza zece la baza opt.

Pentru trecerea de la baza zece la baza șaisprezece considerăm numărul 583 și folosind o așezare ca în figura I.2 avem :

$$(16)\ \begin{array}{c|c|c} 583 & 36 & 2 \\ \hline 7 & 4 & 2 \end{array}\ ; \quad 583 = 247_{(16)}$$

Conversia inversă, de la baza 8 sau 16, la baza 10 se face folosind scrierea

$$N = a_n 10^n_{(8)} + a_{n-1} 10^{n-1}_{(8)} + \ldots + a_1 10_{(8)} + a_0$$

pentru baza 8, iar pentru baza 16

$$N = a_n 10^n_{(16)} + a_{n-1} 10^{n-1}_{(16)} + \ldots + a_1 10_{(16)} + a_0.$$

Trecînd atît cifrele a_i cît și baza $10_{(8)}$ respectiv $10_{(16)}$ în baza 10 se obține pentru baza 8

$$N = a'_n 8^n + a'_{n-1} 8^{n-1} + \ldots + a'_1 8 + a_0$$

iar pentru baza 16

$$N = a''_n 16^n + a''_{n-1} 16^{n-1} + \ldots + a''_1 16 + a''_0,$$

unde cifrele a'_i sînt numerele rezultate din trecerea cifrelor a_i din baza 8 în baza 10 iar a''_i numerele rezultate din trecerea cifrelor a_i din baza 16 în baza 10.

Rezultă pentru trecerea din baza 8 în baza 10

$$N = \sum_{i=0}^{n} a'_i 8^i$$

iar pentru trecerea din baza 16 în baza 10

$$N = \sum_{i=0}^{n} a''_i 16^i.$$

Exemple :

a) Să se facă conversia din octal în zecimal a numărului

$$N = 547_{(8)}.$$

Rezolvare :

$$547_{(8)} = 5 \cdot 8^2 + 4 \cdot 8 + 7 = 359.$$

b) Să se facă conversia din hexazecimal în zecimal a numărului

$$N = \mathrm{FA81}_{(16)}.$$

Rezolvare :

$$\mathrm{FA81}_{(16)} = 15 \cdot 16^3 + 10 \cdot 16^2 + 8 \cdot 16 + 1 = 64\,129.$$

I.7.6. Trecerea părții fracționare a unui număr din sistemul zecimal în sistemul octal și invers

Dacă x este partea fracționară a numărului scris în sistemul zecimal, și vrem să o scriem în octal, folosim scrierea lui x în octal, unde baza $10_{(8)}$ se scrie în baza zece, adică $10_{(8)} = 8$ și avem :

$$x = \frac{a_{-1}}{8} + \frac{a_{-2}}{8^2} + \ldots + \frac{a_{-m}}{8^m}.$$

Înmulțind pe x cu 8 se obține :

$$8x = a_{-1} + \frac{a_{-2}}{8} + \ldots + \frac{a_{-m}}{8^{m-1}}$$

unde se observă că prima cifră de la partea fracționară devine singura cifră a părții întregi a numărului obținut prin înmulțirea cu 8. Repetînd înmulțirea, se poate continua determinarea cifrelor de la partea fracționară a numărului. Procedeul fiind analog cu cel de la baza 2, putem utiliza regula de la I.7.4.1 unde în locul bazei 2 avem baza 8.

De asemenea, facem sublinierea că și în baza 8 putem avea părți fracționare periodice.

Exemplu :

Să se facă conversia din zecimal în octal a numărului 0,484375.

Utilizăm o schemă ca la baza 2 și avem :

(8)	484375	875	0	; $0{,}484375 = 0{,}37_{(8)}$
	0	3	7	

Trecerea inversă, din octal în zecimal, o deducem din :

$$x = \frac{a_{-1}}{10_{(8)}} + \frac{a_{-2}}{10^2_{(8)}} + \ldots + \frac{a_{-m}}{10^m_{(8)}}.$$

Trecînd pe $10_{(8)}$ în baza zece avem $10_{(8)} = 8$ și trecînd și cifrele numărului x din baza 8 în baza 10 deducem

$$x = \frac{a'_{-1}}{8} + \frac{a'_{-2}}{8^2} + \ldots + \frac{a'_{-m}}{8^m},$$

unde a'_i sînt exprimările cifrelor a_i din baza 8 în baza 10, care de fapt coincid.

Deci

$$x = \sum_{i=1}^{m} \frac{a_{-i}}{8^i}.$$

Exemplu :

Să se facă conversia din octal în zecimal pentru numărul : $0{,}176_{(8)}$.

Avem :

$$0{,}176_{(8)} = \frac{1}{8} + \frac{7}{8^2} + \frac{6}{8^3} = \frac{63}{256} = 0{,}24609375.$$

Observație. Cînd numărul asupra căruia se cere conversia are și întregi și parte zecimală, se aplică separat conversia fiecărei părți și apoi se reconstituie numărul din părțile transformate.

I.7.7. Trecerea părții fracționare a unui număr din sistemul zecimal în sistemul hexazecimal și invers

Ca și în celelalte cazuri de trecere a unui număr, parte fracționară, din baza zece în altă bază vom folosi forma numărului scrisă în hexazecimal.

Baza fiind $10_{(16)} = 16$ și numărul fiind x avem :

$$x = \frac{a_{-1}}{16} + \frac{a_{-2}}{16^2} + \ldots + \frac{a_{-m}}{16^m}.$$

Înmulțindu-l pe x cu 16 se obține :

$$16x = a_{-1} + \frac{a_{-2}}{16} + \ldots + \frac{a_{-m}}{16^{m-1}}$$

unde se observă că prima cifră de la partea fracționară devine singura cifră a părții întregi a numărului obținut prin înmulțirea cu 16. Repetînd înmulțirea, se poate continua determinarea cifrelor de la partea fracționară a numărului. Procedeul fiind analog cu cel de la baza 2, putem utiliza regula de la I.7.1.1 unde în locul bazei 2 avem baza 16.

De asemenea, facem sublinierea că în baza 16 putem avea părți fracționare periodice.

Exemplu :

Să se facă conversia din zecimal în hexazecimal a numărului 0,235.

Utilizăm o schemă ca la baza 2, deși se operează mai greu

(16)	235	76	16	56	96	36	76	; $0{,}235 = 0{,}3(C\,28F5)_{(16)}$
	0	3	C	2	8	F	5	

Se observă că după șase înmulțiri succesive ale părții fracționare cu 16, a apărut aceeași parte fracționară care face să se repete cifrele întregilor obținuți prin înmulțire în continuare. Este deci o parte fracționară periodică.

Trecerea inversă, din hexazecimal în zecimal, o deducem din forma numărului scris în hexazecimal

$$x = \frac{a_{-1}}{10_{(16)}} + \frac{a_{-2}}{10^2_{(16)}} + \ldots + \frac{a_{-m}}{10^m_{(16)}}.$$

Trecînd pe $10_{(16)}$ în baza zece avem $10_{(16)} = 16$ și trecînd și cifrele numărului x din baza 16 în baza zece deducem :

$$x = \frac{a'_{-1}}{16} + \frac{a'_{-2}}{16^2} + \ldots + \frac{a'_{-m}}{16^m},$$

unde a'_i sînt exprimările cifrelor a_i în baza zece.

Deci,

$$x = \sum_{i=1}^{m} \frac{a'_{-i}}{16^i}.$$

Exemplu :

Să se facă conversia din hexazecimal în zecimal pentru numărul : $0{,}2F8_{(16)}$.

Avem :

$$0{,}2F8_{(16)} = \frac{2}{16} + \frac{15}{16^2} + \frac{8}{16^3} = \frac{760}{4\,096} = \frac{95}{512}.$$

Observație. Cînd numărul asupra căruia se cere conversia are și întregi și parte zecimală, se aplică separat conversia fiecărei părți și apoi se reconstituie numărul din părțile transformate.

Exemplu :

$$36{,}15 = 36 + 0{,}15 = 24_{(16)} + 0{,}22(6)_{(16)} = 24{,}22(6)_{(16)}.$$

I.7.8. Trecerea numerelor din sistemul binar în sistemul octal și hexazecimal și invers

Pentru această trecere folosim rezultatele de la paragraful I.7.2 și anume regulile de la I.7.2.1 și I.7.2.2 extinse la numere reale.

Deoarece avem $8 = 2^3$, respectiv $16 = 2^4$, urmează că se vor forma grupe de cîte trei, respectiv patru cifre începînd de la virgulă spre stînga și spre dreapta și apoi fiecare grupă de trei, respectiv patru cifre, din baza doi va constitui o cifră în baza opt, respectiv șaisprezece (dacă ultimele grupe formate nu au trei, respectiv patru cifre, se va completa grupa cu zerouri la stînga sau la dreapta după cum împărțirea în grupe s-a făcut la stînga sau la dreapta virgulei).

Pentru ușurință, dăm tabela cu grupele de trei, respectiv patru cifre în binar și echivalentele lor în octal și hexazecimal.

Tabelul I.7

Baza 2	Baza 8	Baza 2	Baza 16	Baza 2	Baza 16
000	0	0000	0	1000	8
001	1	0001	1	1001	9
010	2	0010	2	1010	A
011	3	0011	3	1011	B
100	4	0100	4	1100	C
101	5	0101	5	1101	D
110	6	0110	6	1110	E
111	7	0111	7	1111	F

Exemple :

1. Să se facă conversia din binar în octal și apoi în hexazecimal a numerelor :

a) 1101110101 ; b) 111,011.

Pentru baza 8 formăm grupe de trei cifre la stînga și la dreapta virgulei și cu ajutorul tabelului I.7 în locul fiecărei grupe de trei cifre scriem cifra corespunzătoare în octal.

a) $1101110101_{(2)} = 001.101.110.101_{(2)} = 1565_{(8)}$

b) $111{,}011_{(2)} = 7{,}3_{(8)}$.

Pentru baza 16 formăm grupe de cîte patru cifre și folosind tabelul I.7 ca în cazul bazei 8 avem

a) $110\ 111\ 0101_{(2)} = 0011.0111.0101_{(2)} = 375_{(16)}$

b) $111{,}011_{(2)} = 0111{,}0110_{(2)} = 7{,}6_{(16)}$.

2. Să se facă conversia din octal în binar pentru numerele :

a) $547_{(8)}$; b) $3{,}25_{(8)}$.

Avem :

a) $547_{(8)} = \underbrace{101}_{5}\ \underbrace{100}_{4}\ \underbrace{111}_{7}$ (2)

b) $3{,}25_{(8)} = \underbrace{011}_{3},\ \underbrace{010}_{2}\ \underbrace{101}_{5}$ (2).

3. Să se facă conversia din hexazecimal în binar a numerelor :

a) 7AC ; b) B,0A.

Rezolvare :

a) $7AC_{(16)} = \underbrace{0111}_{7}\ \underbrace{1010}_{A}\ \underbrace{1100}_{C}$ (2) :

b) $B{,}0A_{(16)} = \underbrace{1011}_{B},\ \underbrace{0000}_{0}\ \underbrace{1010}_{A}$ (2).

EXERCIȚII ȘI PROBLEME

1. Fiecare din următoarele operații aritmetice este corectă în cel puțin un sistem de numerație. Să se determine bazele posibile ale acestor sisteme.

a) $1\,234 + 5\,432 = 6\,666$;

b) $\frac{41}{3} = 13$;

c) $\frac{33}{3} = 11$;

d) $23 + 44 + 14 + 32 = 223$;

e) $\frac{302}{20} = 12{,}1$;

f) $\sqrt{41} = 5$;

g) $\sqrt{51} = 6$;

h) $\sqrt{31} = 4$;

i) $\sqrt[3]{11} = 2$;

j) $\sqrt{21} = 3$;

k) $\sqrt{61} = 7$;

l) $\sqrt{a1} = a + 1$, $a \in \mathbf{N}$, $a \geqslant 1$.

2. Se știe că :

a) $16_{(10)} = 100_{(b)}$;
b) $292_{(10)} = 1\,204_{(b)}$.
Să se determine b.

3. Să se arate că :

a) 121 este pătrat perfect în orice bază $b > 2$;
b) 1 331 este cub perfect în orice bază $b > 3$;
c) 14 641 este puterea a 4-a a unui număr, oricare ar fi baza $b > 6$;
d) $(x1)_{(x+2)}$ este pătrat perfect oricare ar fi baza $b = x + 2$.

4. Să se efectueze în baza 2 :

a) 101001 + 110111 + 111010 ;
b) 110001111 − 100011 ;
c) 1101011 − 11011 ;
d) 1111011,01 + 111011,101 + 11,011 ;
e) 10000,001 − 10,011 ;
f) 110110,1011 · 101101,011 ;
g) 11011011,011 : 1,1.

5. Dîndu-se în baza 2 numerele :

$a = 1001{,}11001$;
$b = 1111{,}00011$;
$c = 1010{,}10101$;
să se calculeze :

1° $a + b - c$; 2° $a - b + c$; 3° $-a + b + c$; 4° $\frac{a+b}{c}$; 5° $\frac{bc}{a}$;

6° $\frac{a+b+c}{abc}$.

6. Să se efectueze în baza 16 :

a) ABCDEF + FEDCBA + 12 345 + 54 321 ;
b) BEC − AC + FAC ;
c) AB1 · 7A ;
d) B4 : 6 ;
e) 18A,36 + 0,A72 + D,A5 ;
f) FA,2 · 2,3D.

7. Se dau în baza 16 următoarele numere :

$a = 12A{,}25$;
$b = 108{,}4E$;
$c = FF{,}34$.
Să se calculeze :
1° $4a + 3b - 2c$; 2° $3a - 2b + 4c$; 3° $-2a + 4b + 3c$.

8. Să se treacă următoarele numere dintr-o bază în alta prin intermediul bazei 10.
a) $1\,342_{(5)} = x_{(7)}$; b) $1\,804_{(9)} = x_{(4)}$; c) $32{,}3_{(6)} = x_{(4)}$.

9. Să se facă conversia de la baza 3 la baza 9 a numerelor :
12 101 102 ; 211 022 ; 1 021.

10. Să se facă conversia de la baza 9 la baza 3 a numerelor :
7 568 ; 237,463.

11. Să se facă conversia de la baza 2 la bazele 4, 8 și 16 a numerelor :
1110111101001 ; 110011,01011.

12. Să se facă conversia de la baza 4 la baza 16 a numerelor :
313 312 001 ; 1 230 231,13201.

13. Să se facă conversia de la baza 16 la bazele 8, 4 și 2 a numerelor :
1AB5D8F ; DE4A,19A3.

14. Să se facă conversia de la baza 10 la baza 2 a numerelor :
7 539 432 ; 0,5735 ; 4 726,875.

15. Să se facă conversia de la baza 2 la baza 10 a numerelor :
1001101 ; 0,11101 ; 11,0101.

16. Să se facă conversia de la baza 10 la baza 16 a numerelor :
5 379 ; 0,8475 ; 325,75.

17. Să se facă conversia de la baza 16 la baza 10 a numerelor :
A0FD ; 0,ABC ; 2E,125.

18. Numărul 1 000, scris în baza 10, să fie scris în toate bazele de la 2 la 16 (în afară de zece în care se află deja scris).

Indicații și răspunsuri

1. a) $b \geq 7$; b) 8 ; c) $b \geq 4$; d) 5 ; e) 4 ; f) 6 ; g) 7 ; h) 5 ; i) 7 ; j) 4 ; k) 8 ; l) $a + 2$. **2.** a) $b = 4$; b) $b = 6$. **3.** a) $121 = b^2 + 2b + 1 = (b + 1)^2$; b) ; c) ; d), analog cu a). **4—5.** Se folosesc tabelele I.1 și I.2. **6—7.** Se folosesc tabelele I.5 și I.6. **8.** a) $x_{(7)} = 435_{(7)}$; b) $x_{(4)} = 111\,211_{(4)}$; c) $x_{(4)} = 110{,}2_{(4)}$; **9 — 13.** Se folosesc exemplele de la I.7.8 și regulile de la I.7.2.1 și I.7.2.2. **14—17.** Se folosesc exemplele de la I.7.4 și I.7.5.

II

PRELUCRAREA AUTOMATĂ A DATELOR

II.1. Informația și informatica

Mulțimea și diversitatea informațiilor din lumea actuală („explozia informațională") și dezvoltarea impetuoasă a tehnicii de calcul, au dus la apariția unei noi științe — *informatica*. Termenul de informatică a fost introdus în 1968, fiind sugerat de asocierea informație—automatică. Inițial, informatica a fost definită ca știința prelucrării informației, îndeosebi cu mijloace automate.

Deci obiectul acestei noi științe îl constituie informația. Dar ce este informația ?

S-a încercat definirea informației în diferite moduri, de exemplu :

- fiecare dintre elementele noi, în raport cu cunoștințele prealabile, cuprinse în semnificația unui simbol sau a unui grup de simboluri ;
- formulă scrisă, susceptibilă de a aduce o cunoștință ;
- comunicare, știre, veste ;
- expresie a unui mesaj, prin care se exprimă starea unui fenomen dintr-o anumită activitate, reprezentată printr-un număr, un cuvînt sau un șir de caractere convenționale.

Diferite definiții ale informației reliefează aspecte specifice ale acestei noțiuni. Astfel, distingem : *aspectul semantic*, reflectînd cunoașterea, sens atribuit unei anumite reprezentări ; *aspectul sintactic*, reprezentînd forma căreia i se poate atribui un sens și *aspectul de comunicare*.

Pentru a putea fi percepută, informația trebuie reprezentată pe un suport de informație.

Prin *suport de informație* se înțelege orice mediu material care poate suferi transformări temporare sau permanente, în vederea reprezentării informației. Astfel, creierul uman, moleculele de acizi nucleici, undele sonore și electromagnetice, hîrtia de scris, tabla, banda de magnetofon sînt numai cîteva exemple de suporturi de informație.

În procesul de prelucrare a informațiilor participă *date*, sensul informației fiind exprimat sau extras din acestea.

II.2. Codificarea informațiilor

Informația semantică poate fi considerată ca un ansamblu de trei elemente :

- *entitatea* (obiectul informației) ;
- *atributul* (element de descriere a entității respective, proprietate sau caracteristică a obiectului) ;
- *valoarea* (măsură a atributului).

De exemplu :

a) *manual*	↓ entitate	*prețul în lei* ↓ atribut	*5,20* ↓ valoare
b) *autoturism*	↓ entitate	*marcă* ↓ atribut	*SKODA* ↓ valoare

O entitate poate avea mai multe atribute, fiecăruia corespunzîndu-i mai multe valori. De exemplu, entitatea autoturism poate avea pe lîngă atributul marcă și atributele culoare, proprietar.

Pentru a reprezenta informațiile se recurge la *codificare*.

Un **cod** se definește prin : *alfabet, structură gramaticală* (*sintaxă*) și *atribuire semantică*.

Alfabetul este o mulțime finită de simboluri (semne) numite *caractere* (de exemplu, literele de la *A* la *Z*, cifrele 0, ..., 9).

Sintaxa reprezintă ansamblul regulilor de formare a șirurilor de simboluri care aparțin limbajului de codificare respectiv.

Semantica reprezintă semnificația șirurilor de caractere construite respectînd regulile sintactice.

Construcțiile realizate cu caracterele din alfabet se numesc *cuvinte* (coduri). Cuvintele se constituie în cadrul unui *format*, ceea ce implică limitarea caracterelor utilizabile pentru anumite poziții din cuvînt sau restricții referitoare la ordinea lor. Lungimea cuvintelor poate fi fixă (toate cuvintele au același număr de caractere) sau variabilă.

Dacă formatul limitează lungimea codurilor la n caractere și alfabetul are m caractere distincte, atunci numărul total de coduri distincte care se pot forma este m^n.

Operația inversă codificării se numește *decodificare*.

În prelucrarea automată a datelor se folosesc *coduri binare* (alfabetul este format din două caractere, notate 0 și 1). Codificarea binară a fost impusă de faptul că funcționarea unui calculator se bazează pe elemente cu două stări stabile.

II.3. Arhitectura și structura unui sistem de calcul

Un calculator electronic este un ansamblu de componente electronice, programabil, care poate memora date și efectua cu ele operații (de exemplu, aritmetice și logice).

Pentru a stabili funcțiile pe care trebuie să le îndeplinească un calculator în vederea prelucrării datelor, vom utiliza analogia cu prelucrarea datelor de către om.

Într-adevăr, creierul uman, împreună cu restul sistemului nervos, constituie unul din cele mai complexe sisteme de prelucrare a datelor. El posedă în permanență informații asupra sa și asupra modificărilor care apar în lumea înconjurătoare.

Prin organele de simț recepționăm informații din lumea înconjurătoare. Acestea sînt stocate de memoria noastră ; intervine apoi inteligența care comandă un proces de gîndire prin care se valorifică informațiile, după anumite reguli de prelucrare. Rezultatele prelucrării sînt transmise din nou memoriei. Cunoștințele astfel căpătate le comunicăm lumii exterioare fie prin vorbire, fie prin scriere, acțiuni, gestică. Schematic, acest proces este reprezentat în figura II.1.

În principiu, un calculator electronic trebuie să fie capabil să îndeplinească aceleași funcții ca și calculatorul uman, adică :

- *introducerea datelor* (de exemplu, citirea lor de pe cartele perforate) ;
- *memorarea datelor ;*
- *prelucrarea datelor ;*
- *controlul prelucrării ;*
- *extragerea datelor* (de exemplu, imprimarea rezultatelor sau afișarea lor pe un ecran).

În vederea realizării acestor funcții un sistem de calcul cuprinde echipamente de calcul (hardware) și un sistem de programe (software).

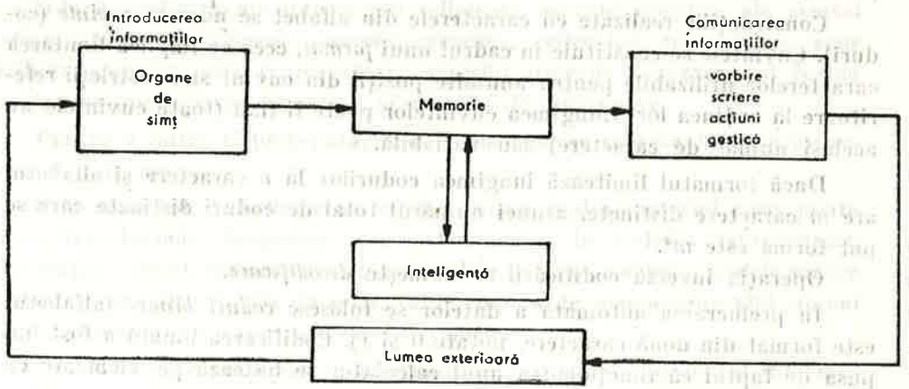


Fig. II.1

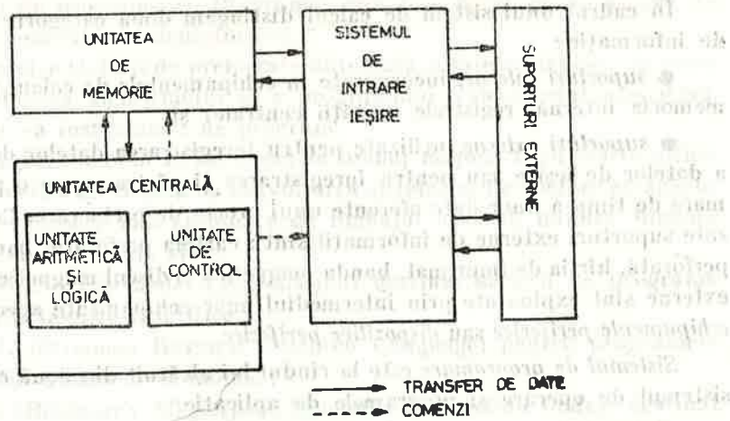


Fig. II.2

Echipamentele de calcul componente sînt descrise în figura II.2.

Unitatea de memorie asigură *păstrarea* datelor (programe, date de prelucrat, rezultate ale prelucrării) care participă într-un proces de prelucrare, *înregistrarea* și *restituirea* datelor la cererea celorlalte unități ale calculatorului. Memoria este constituită dintr-un ansamblu de elemente numite *celule binare.*

Cantitatea de informație ce poate fi înregistrată într-o celulă de memorie se numește *bit*, unui bit fiindu-i asociată valoarea 0 sau 1. Celulele binare sînt grupate în *locații de memorie.* Fiecare locație se identifică printr-un număr, care indică poziția ei în memorie. Acest număr se numește *adresă.* Prin intermediul adresei se realizează *accesul* la informațiile din memorie.

Informația dintr-o locație se numește *octet* sau *bait* (1 octet = 8 biți). 1 024 octeți constituie 1 kilooctet (1 K).

Numărul maxim de octeți ce pot fi înregistrați în memorie determină *capacitatea memoriei.*

Intervalul de timp între emiterea unei cereri de acces la informații din memorie și momentul rezolvării ei se numește *timp de acces* (de obicei mai mic de 1μ sec).

O memorie este cu atît mai performantă cu cît are o capacitate mai mare și un timp de acces mai mic.

Unitatea centrală asigură prelucrarea propriu-zisă și comandă funcționarea celorlalte componente. Unitatea centrală dispune și de un *panou de comandă,* constituit dintr-un ansamblu de beculețe și comutatoare, care permite dialogul dintre operatorul la calculator și sistem. O parte din aceste funcții pot fi realizate și cu ajutorul altei componente — *consola operator.*

Sistemul de intrare-ieșire asigură schimbul de informații între mediul exterior și calculator.

În cadrul unui sistem de calcul distingem două categorii de suporturi de informație :

● *suporturi interne*, încorporate în echipamentele de calcul (de exemplu, memoria internă, registrele unității centrale) și

● *suporturi externe*, utilizate pentru înregistrarea datelor de intrare sau a datelor de ieșire sau pentru înregistrarea și păstrarea pe o perioadă mai mare de timp a unor date aferente unui proces de prelucrare. Cele mai utilizate suporturi externe de informații sînt : cartela perforată, banda de hîrtie perforată, hîrtia de imprimat, banda magnetică și discul magnetic. Suporturile externe sînt exploatate prin intermediul unor echipamente speciale, numite *echipamente periferice* sau *dispozitive periferice*.

Sistemul de programare este la rîndul lui alcătuit din două componente : sistemul de operare și programele de aplicație.

Sistemul de operare cuprinde o mulțime de programe avînd rolul de a mări performanțele calculatorului prin furnizarea de facilități care nu sînt prevăzute de hardware. De exemplu, asigură încărcarea automată a programelor în memorie, traducerea în limbaj mașină a programelor scrise în limbaje simbolice, controlează execuția programelor și detectează automat unele tipuri de erori etc.

Programele de aplicații sînt programele utilizatorilor. În cadrul acestora o categorie deosebită o constituie programele de aplicații standard, de tipul programelor pentru rezolvarea sistemelor de ecuații, pentru calculul cu matrice, programele de gestiune etc., care sînt înregistrate pe un suport extern de informație și puse la dispoziția oricărui utilizator.

II.4. Limbaje de programare

Calculatorul poate rezolva orice problemă dacă i se specifică *cum* să o rezolve.

O problemă poate fi rezolvată cu calculatorul dacă s-a găsit o schemă de rezolvare, respectiv un algoritm.

Algoritmul este o succesiune de indicații univoc prezentate, care descriu modul de rezolvare a unor probleme de același tip, descrierea fiind exactă și completă și cuprinzînd un număr finit de pași.

O dată stabilit un algoritm, el este de obicei reprezentat grafic sub forma unei *scheme logice*, pentru a putea fi mai ușor de urmărit. Plecînd de la această reprezentare grafică se scrie apoi *programul* (reprezentarea algoritmului

într-un limbaj de programare). Programul este înregistrat pe un suport de informație și comunicat calculatorului.

Reușita unei activități de prelucrare automată a datelor depinde în mare măsură de calitatea algoritmului și a programului, calculatorul executînd numai ceea ce i-a fost indicat de program.

Inițial programele erau scrise direct în limbaj mașină, fiind foarte dificil de elaborat și verificat. O dată cu dezvoltarea sistemelor de operare au apărut limbaje de programare mai apropiate de limbajul natural, numite limbaje evoluate. Astfel de limbaje sînt :

● FORTRAN (FORmula TRANslation) destinat scrierii de programe pentru aplicații tehnico-științifice ;

● COBOL (COmmon Business Oriented Language) pentru programele de aplicații din domeniul economic ;

● BASIC (Beginner's All-purpose Symbolic Instruction Code), destinat unui domeniu larg de aplicații.

În afară de limbajele evoluate, există și o categorie intermediară — limbajele de asamblare, care de fapt reprezintă o exprimare prin simboluri a limbajului mașină. Exemple de limbaje de asamblare sînt : ASSIRIS (pentru calculatoarele FELIX), MACRO (pentru minicalculatoarele FELIX-M).

În general, limbajele evoluate sînt independente de sistemul de calcul, în timp ce limbajele de asamblare sînt specifice unui anumit sistem de calcul.

II.5. Dezvoltare de programe

Programele sînt scrise în limbaje simbolice, ușor accesibile programatorului. Pentru a putea fi executate, ele vor trebui traduse în limbaj mașină. Transformarea programelor într-o formă acceptată de calculator este preluată de sistemul de operare. Această operație se realizează de obicei în două faze, prin execuția unor programe ale sistemului de operare : programul *compilator* pentru limbajul în care este scris programul utilizatorului și *editorul de legături* care transformă programul compilat într-un program direct executabil. Programul direct executabil este încărcat în memorie și apoi lansat în execuție (fig. II.3).

Deci fazele de dezvoltare ale unui grup sînt :

● compilarea ;

● editarea de legături ;

● execuția propriu-zisă.

Fazele de dezvoltare se comunică sistemului prin comenzi speciale.

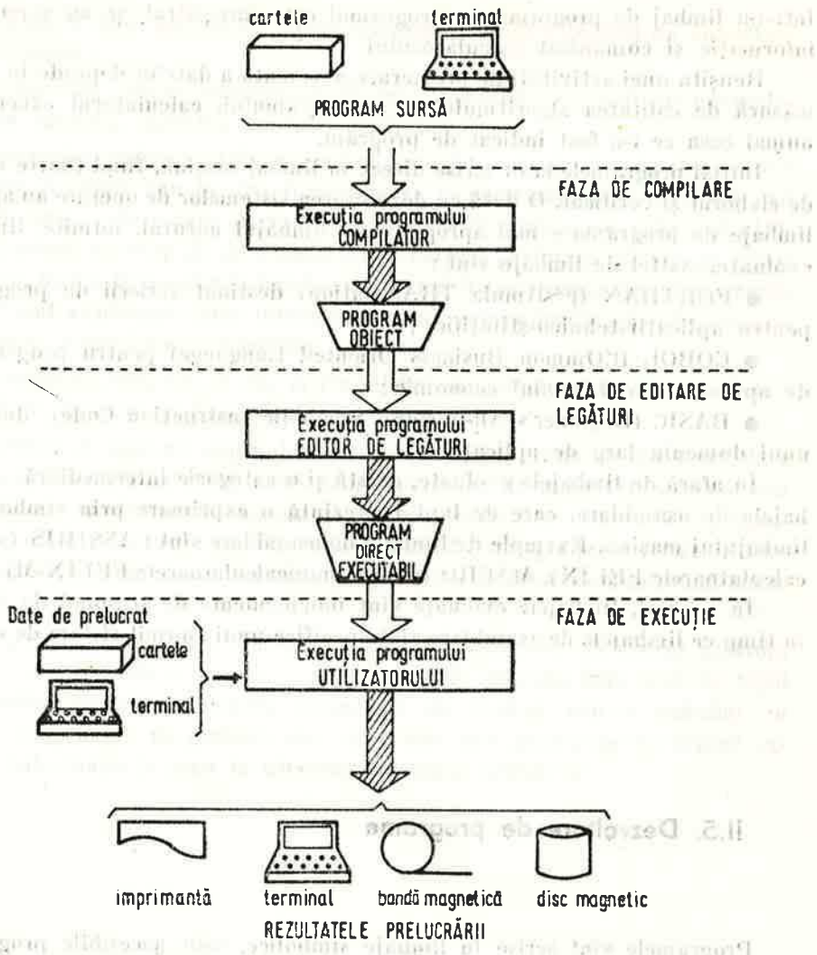


Fig. II.3

II.6. Informatica în România

La noi în țară preocupările în domeniul tehnicii de calcul datează din 1950.

În 1957, la Institutul de Fizică Atomică s-a construit primul calculator electronic românesc CIFA-1. Acesta era un calculator din generația I* având o viteză de calcul de 50 operații/secundă.

* În tabelul II.1 se face o trecere succintă în revistă a generațiilor de calculatoare și a principalelor caracteristici.

Tabelul II.1

Generații de calculatoare

Generația	Anul apariției	Tehnologie	Tip de memorie internă	Facilități oferite utilizatorilor
1	1950	• relee • tuburi electronice	• tuburi catodice • linii de întîrziere	• programare în cod mașină • apare „programul memorat"
2	1955	• tranzistori • semiconductori	• ferite	• apar limbaje mai evoluate și deci compilatoare pentru aceste limbaje • transferul datelor este preluat de unități speciale — unități de schimburi (canale)
3 FELIX C-256 FELIX C-512 FELIX C-1024	1968	• circuite integrate	• inele de ferită	• limbaje evoluate • sisteme de operare pentru familii de calculatoare • multiprogramare • multiprelucrare • prelucrări în timp real
3,5 CORAL INDEPENDENT M-18 M-118 ECAROM	1973	• circuite integrate (MSI, LSI)	• circuite integrate	• memorii foarte rapide • ridicarea restricțiilor legate de dimensiunea memoriei interne • software pentru baze de date • prelucrare distribuită

La Timişoara s-au realizat calculatoare din seria MECIPT, iar la Cluj-Napoca, DACICC.

O deosebită importanţă în promovarea informaticii în ţara noastră au avut-o cursurile de calculatoare organizate la Bucureşti, încă din 1957, de Gr. C. Moisil şi la Cluj-Napoca de Tiberiu Popoviciu, activitate continuată cu succes de un număr din ce în ce mai mare de discipoli.

În 1967, conducerea partidului a aprobat un program de dotare cu tehnică de calcul şi automatizare a prelucrării datelor. S-au înfiinţat centre teritoriale de calcul electronic (CTCE) precum şi centre de calcul în cadrul unor întreprinderi. De asemenea, s-au pus bazele unei industrii electronice proprii, trecîndu-se la realizarea de componente electronice pentru tehnică de calcul. O atenţie deosebită s-a acordat pregătirii de cadre de specialitate, prin instituţii de învăţămînt superior, centre de perfecţionare şi unele licee.

Din 1968 a început fabricarea în ţara noastră a unor calculatoare electronice din generaţia a 3-a, FELIX C-256.

Alături de acestea, s-au realizat prin concepţie proprie noi calculatoare din familia FELIX : C-512, C-1024, C-32.

În prezent, producem sisteme de calcul cu o arhitectură modernă, bazîndu-ne pe tehnologii avansate : familia de minicalculatoare FELIX-M (INDEPENDENT I-100, I-102F şi CORAL 4001, 4011, 4030), microcalculatoarele M-18, M-118, ECAROM, CUB, JUNIOR şi calculatoarele profesionale şi individuale : FELIX-PC, HC-85, aMIC, PRAE şi TIM-S. Apariţia calculatoarelor profesionale şi individuale va facilita răspîndirea tehnicii de calcul în mediile neindustriale şi va permite familiarizarea elevilor, încă din ciclul gimnazial, cu calculatorul electronic.

III

OPERAŢII ARITMETICE ŞI OPERAŢII LOGICE

III.1. Noţiuni introductive

Din şcoala generală cunoaştem că mulţimea numerelor reale $\mathbf{R}$ este formată din reuniunea a două submulţimi disjuncte, mulţimea numerelor raţionale $\mathbf{Q}$ şi mulţimea numerelor iraţionale $\mathbf{R} \setminus \mathbf{Q}$.

De asemenea, cunoaştem din capitolul I că orice număr real se poate scrie într-o bază oarecare, deci şi în baza zece, ca o succesiune de cifre la partea întreagă şi o succesiune de cifre la partea zecimală. În practică, în calcule, nu totdeauna utilizăm sau putem utiliza un anumit număr real, mai ales în cazul numerelor iraţionale. De aceea ori de cîte ori tehnica de lucru se uşurează, fără ca rezultatele să devină inutilizabile, se vor putea folosi, în locul oricăror numere reale, valori apropiate ale acestora, numite valori aproximative sau, simplu, aproximante.

III.2. Aproximantele numerelor reale

Noţiunea de aproximantă (valoare care aproximează un anumit număr) o vom utiliza pentru orice număr real care urmează a fi folosit în calcule.

De aceea, considerînd x un număr real oarecare ($x \in \mathbf{R}$), pentru aproximantele lui x dăm următoarea :

III.2.1. *Definiţie.* **Numim aproximantă a unui număr real x orice număr raţional x^* care poate fi folosit în calcule în locul numărului x.**

Exemplu :

Dacă $x = 24{,}235$, atunci numerele 20 ; 24 ; 25 ; 24,2 ; 24,3 ; 24,23 ; 24,24 ; 10 ; 100 ; 100 000 şi oricare altele pot fi considerate ca aproximante ale lui x.

Mulţimea aproximantelor lui x o vom aprecia ca avînd două submulţimi, o submulţime formată cu aproximantele mai mici decît x, ale cărei elemente le vom numi aproximante prin lipsă, notate cu $\underline{x}$, şi o submulţime formată cu aproximantele mai mari decît x ale cărei elemente le vom numi aproximante prin adaos sau aproximante prin exces, notate cu $\bar{x}$.

A considera pentru un număr real x o aproximantă x^*, înseamnă a aproxima numărul real x cu aproximanta x^*.

III.3. Procedee de aproximare a numerelor reale

III.3.1. Aproximări prin rotunjire la numerele cu parte zecimală (regula completării)

Fie un număr real x care are cel puțin n cifre la partea zecimală. Dorim să-l aproximăm cu o aproximantă a sa care să aibă n cifre la partea zecimală. Putem fi în unul din următoarele două cazuri:

a) cifra $a_{-(n+1)}$ este una dintre cifrele 0, 1, 2, 3, 4, și în acest caz se renunță la această cifră precum și la cele de la dreapta ei;

b) cifra $a_{-(n+1)}$ este una din cifrele 5, 6, 7, 8, 9 și în acest caz adunăm la număr $\frac{1}{10^n}$ și renunțăm la ultimele cifre zecimale, începînd cu poziția a $n+1$-a.

Spunem că atît în cazul a) cît și în cazul b) am efectuat o aproximare prin rotunjire.

III.3.1.1. *Regulă.* Aproximarea prin rotunjire a unui număr real cu cel puțin n zecimale, se face prin lipsă sau prin exces la zecimala a n-a, renunțînd la zecimalele care-i urmează, zecimala a n-a rămînînd neschimbată sau majorată cu o unitate după cum zecimala care-i urmează este o cifră mai mică decît cinci sau mai mare ori egală cu cinci.

III.3.1.2. *Observații:*

1. Aproximanta obținută prin rotunjire la III.3.1. a) este o aproximantă prin lipsă cînd numărul real este pozitiv.

2. Aproximanta obținută prin rotunjire la III.3.1. b) este o aproximantă prin exces cînd numărul real este pozitiv.

Exemple:

1) Numărul $x = 2{,}8435$ dorim să fie scris doar cu două zecimale, aproximat prin rotunjire, deci $n = 2$. Deoarece $a_3 = 3$ și $3 < 5$, atunci $\underline{x} = 2{,}84$ reprezintă o aproximantă prin lipsă.

2) Numărul $x = 324{,}76583$ dorim să fie scris doar cu trei zecimale, aproximat prin rotunjire, deci $n = 3$. Deoarece, $a_4 = 8$ și $8 > 5$, atunci $\bar{x} = 324{,}766$ reprezintă o aproximantă prin exces.

III.3.2. Aproximări prin rotunjire la numerele întregi (regula completării)

Se dă un număr întreg x cu cel puțin atîtea cifre cît este ordinul la care vrem să facem aproximarea. Dorim să-l aproximăm la una dintre cifrele: unităților, zecilor, sutelor, miilor etc. Notăm cu k acest ordin și cu $k-1$ ordinul imediat inferior. Procedăm ca în unul din următoarele două cazuri:

a) înlocuim cu zerouri toate cifrele de la dreapta cifrei de ordinul k, dacă cifra de ordin $k-1$ este una dintre cifrele 0, 1, 2, 3, 4;

b) adunăm 10^k cu numărul dat și înlocuim toate cifrele de la dreapta cifrei de ordinul k prin zerouri cînd cifra de ordin $k-1$ este una dintre cifrele 5, 6, 7, 8, 9.

Spunem că atît în cazul a) cît și în cazul b) am efectuat o aproximare prin rotunjire.

III.3.2.1. *Regulă.* Aproximarea prin rotunjire a unui număr întreg la o cifră de un anumit ordin fixat se face prin lipsă sau prin exces, înlocuind prin zerouri cifrele care urmează cînd prima cifră care-i succede este mai mică decît 5, sau majorînd cifra fixată cu o unitate și înlocuind toate cifrele care-i urmează cu zerouri, dacă prima cifră care-i succede este mai mare ori egală cu cinci.

III.3.2.2. *Observații:*

1. Aproximanta obținută prin rotunjire la III.3.2. a) este o aproximantă prin lipsă cînd $x > 0$.

2. Aproximanta obținută prin rotunjire la III.3.2. b) este o aproximantă prin exces cînd $x > 0$.

Notînd cu c cifra de ordin k și cu d cifra de ordin $k-1$ în următoarele exemple avem:

1. Numărul $x = 7589345$ dorim să-l aproximăm la cifra miilor prin rotunjire, deci $c = 9$. Pentru că $d = 3$ și $3 < 5$, atunci $\underline{x} = 7589000$ reprezintă o aproximantă prin lipsă.

2. Numărul $x = 47979463$ dorim să-l aproximăm la cifra zecilor de mii prin rotunjire, deci $c = 7$. Pentru că $d = 9$ și $9 > 5$, atunci $\bar{x} = 47980000$ reprezintă o aproximantă prin exces.

3. Numărul $x = -32456$ dorim să-l aproximăm la cifra zecilor prin rotunjire, deci $c = 5$. Pentru că $d = 6$ și $6 > 5$, atunci $\underline{x} = -32460$ reprezintă o aproximantă prin lipsă.

4. Numărul $x = -83459478$ dorim să-l aproximăm la cifra miilor prin rotunjire, deci $c = 9$. Pentru că $d = 4$ și $4 < 5$, atunci $\bar{x} = -83459000$ reprezintă o aproximantă prin exces.

5. Numărul $x = 573499953$ dorim să-l aproximăm la cifra sutelor prin rotunjire, deci $c = 9$. Pentru că $d = 5$, atunci $\bar{x} = 573500000$.

6. Numărul $x = 15700000$ dorim să-l aproximăm la cifra zecilor de mii sau la orice cifră de la dreapta ei. În acest caz $\bar{x} = \underline{x} = x$.

III.3.3. Aproximarea prin rotunjire efectuată la mașinile de calcul

În aproximarea prin rotunjire este necesară comparația cifrei la care se face rotunjirea cu cifra 5 precum și cercetarea semnului numărului pentru a se putea decide în ce mod se face rotunjirea (prin lipsă sau prin exces).

Deoarece, pentru mașinile de calcul această operație este complicată și poate duce uneori la un consum mare de timp, se înlocuiește operația de comparare cu o operație de adunare.

Astfel, în cazul în care x este un număr real scris sub formă de fracție zecimală cu cel puțin n zecimale și dorim să-l scriem printr-o aproximantă cu n zecimale, adunăm la acest număr numărul cu același număr de zecimale și același semn cu el avînd toate zecimalele zerouri în afară de zecimala a $n+1$-a care este 5. Rezultatului astfel obținut i se neglijează cifrele după zecimala a n-a. Aproximanta obținută este o aproximantă prin rotunjire.

Exemple :

1) Numărul $x = 68{,}975342$ dorim să fie scris cu patru zecimale, deci $n = 4$. Îi adunăm numărul $0{,}000050$ și avem $68{,}975342 + 0{,}000050 = 68{,}975392$. La rezultatul astfel obținut se neglijează zecimalele de după zecimala a 4-a, obținîndu-se $x = 68{,}9753$, rezultat identic cu cel ce s-ar fi obținut aplicînd rotunjirea.

2) Numărul $x = -732{,}84526$ dorim să fie scris cu două zecimale, deci $n = 2$. Îi adunăm numărul $-0{,}00500$ și avem $-732{,}84526 - 0{,}00500 = -732{,}85026$. La rezultatul astfel obținut se neglijează zecimala de după zecimala a 2-a, obținîndu-se $x = -732{,}85$, rezultat identic cu cel ce s-ar fi obținut aplicînd rotunjirea.

Fie acum cazul în care x este un număr întreg cu n cifre și dorim să-l rotunjim la o cifră de un anumit ordin k și care este una din cifrele : unități, zeci, sute, mii etc. Pentru aceasta adunăm la numărul dat numărul de același semn cu el, cu un număr de cifre egal cu numărul de cifre de la dreapta cifrei de ordin k, avînd prima cifră 5, iar următoarele cifre zerouri. Rezultatului astfel obținut i se substituie cifrele de după cifra de ordin k prin zerouri.

Exemple :

1) Numărul $x = 97\,354\,942$ dorim să-l aproximăm la cifra zecilor de mii, deci la a 5-a cifră de la dreapta la stînga. Îi adunăm numărul de patru cifre de același semn avînd prima cifră 5 și celelalte zerouri, și avem $97\,354\,942 + 5\,000 = 97\,359\,942$. La rezultatul obținut se substituie ultimele 4 cifre cu zerouri. Obținem : $x = 97\,350\,000$.

2) Numărul $x = -475\,394\,278$ dorim să-l aproximăm la ordinul zecilor de milioane, deci la a 8-a cifră de la dreapta la stînga. Îi adunăm numărul de 7 cifre de același semn avînd prima cifră de 5 și celelalte zerouri, și avem $-475\,394\,278 - 5\,000\,000 = -480\,394\,278$. La rezultatul astfel obținut se substituie ultimele 7 cifre cu zerouri. Se obține $x = -480\,000\,000$.

III.3.4. Aproximarea canonică sau prin trunchiere

De foarte multe ori, mai ales în operații cu numere, cînd ne interesează aproximante ale rezultatelor, sîntem conduși a considera aproximante care să rețină numai un anumit număr de zecimale sau cifrele de la un anumit ordin în sus. Spunem că în acest caz facem o aproximare canonică sau prin trunchiere.

Fie un număr real x scris sub formă de fracție zecimală cu cel puțin n zecimale, pe care vrem să-l aproximăm folosind forma canonică reținînd n

zecimale. Pentru aceasta se renunță la toate zecimalele de după zecimala a n-a, obținîndu-se o aproximantă a numărului x.

III.3.4.1. *Regulă.* Pentru aproximarea canonică a unui număr real x scris sub formă de fracție zecimală cu cel puțin n zecimale din care vrem să reținem primele p zecimale ($p \leqslant n$) vom renunța la ultimele $n - p$ zecimale.

Exemple :

1) Numărul $x = 0{,}4597323$ dorim să-l aproximăm, folosind aproximația canonică, reținînd primele 4 zecimale. Deci $x = 0{,}4597$.

2) Numărul $x = -23{,}579432$ dorim să-l aproximăm, folosind aproximația canonică, reținînd primele două zecimale. Deci $\bar{x} = -23{,}57$.

Pentru un număr întreg x cu cel puțin n cifre, aproximarea canonică de un anumit ordin înseamnă a înlocui cifrele care se află la dreapta cifrei de acel ordin cu zerouri, obținîndu-se astfel o aproximantă a numărului x.

III.3.4.2. *Regulă.* Pentru aproximarea canonică a unui număr întreg x cu cel puțin n cifre, din care vrem să reținem primele $p (p \leqslant n)$ cifre (de la stînga) ale numărului, substituim ultimele $n - p$ cifre (de la dreapta) ale numărului cu zerouri.

Exemplu :

Numărul $x = 13\,984\,572$ dorim să-l aproximăm, folosind aproximația canonică, reținînd primele patru cifre (de la stînga). Avem $\underline{x} = 13\,980\,000$, care este o aproximantă a lui x.

III.4. Erori și tipuri de erori

III.4.1. Noțiunea de eroare

Cînd înlocuim un număr real x cu altul x^*, care se numește aproximantă a numărului x, atunci această aproximantă x^* exprimă pe x cu un anumit grad de exactitate sau precizie. Evident, cu cît x^* este mai apropiat de valoarea lui x, cu atît x^* aproximează mai bine pe x.

Pentru măsurarea gradului de precizie cu care o aproximantă x^* aproximează pe x vom căuta să știm care este diferența dintre aproximanta x^* și numărul x pe care îl aproximează și această diferență o vom numi eroare.

III.4.1.1. *Definiție.* **Se numește eroare a unei aproximante față de numărul real aproximat, diferența între aproximantă și numărul real pe care îl aproximează.**

Notînd numărul cu x, aproximanta sa cu x^* și eroarea aproximantei cu ε vom avea, potrivit definiției, relația :

$$\varepsilon = x^* - x.$$

Deoarece aproximanta unui număr real poate fi prin lipsă sau prin exces, putem avea eroarea dată de una din următoarele două relații :

$$\varepsilon = \underline{x} - x \text{ sau } \varepsilon = \bar{x} - x.$$

Însă din $\underline{x} < x$ rezultă $\underline{x} - x < 0$, adică $\varepsilon < 0$, iar din $\bar{x} > x$ rezultă $\bar{x} - x > 0$, adică $\varepsilon > 0$, ceea ce înseamnă că eroarea unei aproximante poate fi negativă sau pozitivă.

III.4.1.2. *Definiție.* **Spunem că aproximanta x^* a unui număr real x este prin exces sau prin lipsă după cum eroarea aproximantei x^* este pozitivă sau negativă.**

Exemplu :

Fie numărul $x = 279{,}453$ și două aproximante ale sale $\underline{x} = 279{,}45$ și $\bar{x} = 279{,}5$. Erorile ce corespund celor două aproximante sînt :

$$\varepsilon = -0{,}003 \text{ și } \varepsilon = 0{,}047.$$

III.4.2. Eroare absolută, eroare absolută maximă

În definirea noțiunii de eroare am constatat că eroarea poate fi atît pozitivă cît și negativă. Deoarece în foarte multe cazuri nu are importanță semnul erorii, ci interesează mărimea acesteia, vom defini noțiunea de eroare absolută.

III.4.2.1. *Definiție.* **Eroarea absolută a unei aproximante este valoarea absolută a erorii sale.**

Potrivit definiției, pentru un număr real x și o aproximantă a sa x^* eroarea absolută va fi $|x^* - x|$. Vom nota eroarea absolută cu a unde $a > 0$. Avem deci relația :

$$a = |x^* - x|.$$

Cunoașterea erorii absolute a unei aproximante ne permite să dăm o încadrare a numărului real :

$$(|x^* - x| = a) \Leftrightarrow (x^* - a \leqslant x \leqslant x^* + a).$$

Noțiunea de eroare absolută ne permite să facem aprecieri asupra aproximării numărului de către o aproximantă sau alta. Astfel dacă x_1^* și x_2^* sînt două aproximante ale aceluiași număr real x și dacă $a_1 < a_2$ spunem că x_1^* aproximează mai bine pe x decît x_2^* și deci x_1^* este o aproximantă mai bună a lui x decît x_2^*.

Exemplu :

Fie numărul $x = 1{,}45724$ cu aproximantele $x_1^* = 1{,}457$ și $x_2^* = 1{,}46$ cu erorile absolute $a_1 = 0{,}00024$ și $a_2 = 0{,}00276$. Deoarece $a_1 < a_2$ aproximanta x_1^* aproximează mai bine pe x decît x_2^* sau, ceea ce este totuna, x_2^* aproximează mai slab pe x decît aproximanta x_1^*.

Este de remarcat că media aritmetică a două aproximante ale aceluiași număr este o aproximantă mai bună decît cel puțin una din cele două.

Exemplu :

Fie numărul $x = 21{,}548$ cu aproximantele $x_1^* = 21{,}54$ și $x_2^* = 21{,}55$. Erorile fiind date de $\varepsilon_1 = -0{,}008$ și $\varepsilon_2 = 0{,}002$, media aritmetică a aproximantelor este $x^* = \frac{21{,}54 + 21{,}55}{2} =$ $= 21{,}545$ iar eroarea pentru noua aproximantă este $\varepsilon = 0{,}003$. Deci noua aproximantă x^* este mai bună decît x_1^* dar mai slabă decît x_2^*.

În operațiile aritmetice cu aproximante, în foarte multe cazuri, sîntem în situația că diferite operații conduc la erori diferite, ceea ce face ca în general să ne punem problema cunoașterii sau determinării celei mai mari erori pentru a putea estima nivelul preciziei cu care se lucrează. De aceea vom introduce noțiunea de eroare absolută maximă prin următoarea

III.4.2.2. *Definiție.* **Numim eroare absolută maximă pentru toate aproximantele x_i^* pentru numărul real x, cel mai mic număr A care satisface inegalitatea $A \geqslant \max(a_i)$, unde a_i sînt erorile absolute ale aproximantelor x_i^*.**

Este evident că pentru un singur număr x și pentru o singură aproximantă a sa x^* eroarea absolută maximă este chiar eroarea absolută a acelei aproximante, deoarece $a \leqslant a$ pentru orice a. De aceea vom utiliza noțiunea de *eroare absolută maximă a unei aproximante* și în cazul unei singure aproximante.

Eroarea absolută este mai mică sau egală cu numărul ce se obține prin înlocuirea în x a cifrei ordinului de aproximare cu 1 și a tuturor celorlalte cifre cu zero pentru orice fel de aproximare la cifra de ordinul respectiv. Urmează că putem considera numărul astfel obținut ca eroare absolută maximă a aproximantelor prin lipsă sau exces ale numărului x pentru cifra de ordinul fixat.

III.4.3. Eroare relativă, eroare relativă maximă

Cunoașterea numai a erorii absolute a unei aproximante pentru un număr real este de foarte multe ori insuficientă pentru a caracteriza gradul de precizie într-un calcul sau într-o măsurătoare. Așa, de exemplu, dacă afirmăm că eroarea unei anumite lungimi măsurate este de 1 mm, fără a se cunoaște și mărimea lungimii măsurate, nu se poate spune dacă măsurarea este bine făcută sau nu. Dacă eroarea de 1 mm se referă la măsurarea unei lungimi de 50 m se poate spune că rezultatul este extraordinar de bun, dar dacă s-ar măsura diametrul interior al unui rulment și în loc de 10 mm am aprecia prin măsurătoare că are 9 mm, este evident că nimeni nu poate aprecia rezultatul ca fiind bun.

De aceea, calitatea unui rezultat poate fi cu mult mai bine apreciată, dacă se cunoaște așa-numita eroare relativă în definirea căreia intră și mărimea însăși.

III.4.3.1. *Definiție.* **Eroarea relativă a unei aproximante x^* pentru un număr real x este egală cu raportul dintre eroarea absolută a aproximantei x^* și valoarea absolută a numărului x.**

Numărul real fiind x, aproximanta sa x^* și notînd eroarea relativă cu e vom avea, potrivit definiției

$$e = \frac{|x^* - x|}{|x|} = \frac{a}{|x|}.$$

Exemplu:

Fie $x = 1{,}415$ și $x^* = 1{,}42$ avem $\varepsilon = 0{,}005$, iar $e = \frac{|0{,}005|}{|1{,}415|}$, care este aproximativ 0,004.

Deseori eroarea relativă se exprimă fie la sută (în procente), fie la mie. În exemplul dat, eroarea relativă este de $4\,^0/_{00}$ (4 la mie).

Ca și pentru eroarea absolută maximă și aici considerente de ordin practic ne determină să introducem noțiunea de eroare relativă maximă.

Dăm astfel următoarea

III.4.3.2. *Definiție.* **Numim eroare relativă maximă a aproximantelor x_i^* pentru numărul real x, pe care o notăm cu E, raportul dintre eroarea absolută maximă și valoarea absolută a numărului.**

Pentru un singur număr x și pentru o singură aproximantă a sa x^* eroarea relativă maximă este chiar eroarea relativă a acelei aproximante, deoarece eroarea absolută maximă, în acest caz, este chiar eroarea absolută a aproximantei. De aceea vom utiliza noțiunea de *eroare relativă maximă a unei aproximante* și în cazul unei singure aproximante.

Potrivit definiției date și a notațiilor adoptate avem:

$$E = \frac{A}{|x|}.$$

III.5. Erori datorate modului de reprezentare

În operațiile aritmetice cu numere reale sîntem adeseori conduși să operăm cu aproximante care au un număr finit de n zecimale, în general mai mic decît numărul de zecimale η, al numerelor reale date ($\eta \geqslant n$), fără ca rezultatul obținut să devină inutilizabil.

Deoarece în locul numerelor reale date operăm cu aproximantele lor care au altă reprezentare (reprezentare finită, cu un număr dat de n zecimale sau cifre), eroarea introdusă se numește *eroare de reprezentare.*

Eroarea de reprezentare este prezentă de la început în calcule și străbate tot complexul de operații pătrunzînd în rezultat. De aceea vom spune că eroarea de reprezentare este o eroare transmisibilă.

III.6. Inegalitățile fundamentale în aproximarea numerelor reale

Fie numărul real pozitiv x dat prin

$$x = a_n a_{n-1} \ldots a_1 a_0, a_{-1} a_{-2} \ldots a_{-k} \ldots$$

unde $a_n \neq 0$ (a_n este prima cifră diferită de zero în scrierea numărului x de la stînga la dreapta)

și

$$x^* = a_n a_{n-1} \ldots a_1 a_0, a_{-1} a_{-2} \ldots a_{-k}$$

o aproximantă a lui x. Putem scrie această aproximantă folosind forma de scriere cu ajutorul puterilor lui 10.

$$x^* = a_n 10^n + a_{n-1} 10^{n-1} + \ldots + a_1 10 + a_0 + \frac{a_{-1}}{10} + \frac{a_{-2}}{10^2} + \ldots + \frac{a_{-k}}{10^k}.$$

Nu este greu de remarcat că atît x cît și x^* admit încadrarea

$$10^n \leqslant x < 10^{n+1} \text{ și } 10^n \leqslant x^* \leqslant 10^{n+1}.$$

Pentru demonstrație vom considera mai întîi o aproximantă $\bar{x}$ prin adaos a lui x în care partea zecimală se substituie cu 1 obținînd $\bar{x} - 1 \leqslant x \leqslant \bar{x}$.

Ținînd seama că pentru cifrele de la partea stîngă a aproximantei avem relațiile

$$1 \leqslant a_n \leqslant 9$$

$$0 \leqslant a_k \leqslant 9 \qquad \text{pentru } k = 1, 2, \ldots, n-1.$$

$$1 \leqslant a_0 + 1 \leqslant 10$$

înmulțind prima inegalitate cu 10^n și a doua cu 10^k, apoi sumînd avem

$$10^n + 1 \leqslant \bar{x} \leqslant 9(10^n + 10^{n-1} + \ldots + 10) + 10.$$

Efectuînd suma din partea dreaptă a celei de a doua inegalități rezultă

$$10^n + 1 \leqslant \bar{x} \leqslant 10^{n+1}$$

care se poate scrie

$$10^n \leqslant \bar{x} - 1 \text{ și } \bar{x} \leqslant 10^{n+1},$$

unde folosind și $\bar{x} - 1 \leqslant x$, $x \leqslant \bar{x}$ scrise anterior avem

$$10^n \leqslant \bar{x} - 1 \leqslant x \leqslant \bar{x} \leqslant 10^{n+1}$$

care numai pentru numărul x devine

$$10^n \leqslant x < 10^{n+1}.$$

Inegalitățile pentru aproximanta x^* se pot scrie direct, dacă avem în vedere că demonstrația pentru $\bar{x}$ poate avea loc indiferent de ordinul la care se face rotunjirea prin adaos.

Deci

$$10^n \leqslant x^* \leqslant 10^{n+1}.$$

Dubla relație obținută pentru x formează așa-numita *prima dublă inegalitate* fundamentală în aproximarea numerelor reale iar exponentul n care apare în puterile lui 10 este numit caracteristica numărului real x.

De asemenea numărul x și aproximanta lui x^*, dată anterior, satisfac și relațiile :

$$a_n 10^n \leqslant x < (a_n + 1)10^n, \quad a_n \cdot 10^n \leqslant x^* \leqslant (a_n + 1)10^n.$$

Pentru a demonstra aceste inegalități vom considera aproximanta $\bar{x}$ de mai înainte formată din două părți. Astfel :

$$\bar{x} = a_n \cdot 10^n + N.$$

Ținînd seama că pentru orice cifră din N avem

$$0 \leqslant a_k \leqslant 9 \qquad \text{pentru } k = 1, 2, \ldots, n-1$$

$$1 \leqslant a_0 + 1 \leqslant 10.$$

Înmulțind prima relație cu 10^k și sumînd avem

$$1 \leqslant N \leqslant 9(10^{n-1} + 10^{n-2} + \ldots + 10) + 10.$$

Făcînd suma în partea dreaptă a inegalității obținem

$$1 \leqslant N \leqslant 10^n.$$

Adunînd acum la fiecare parte $a_n 10^n$ găsim

$$a_n 10^n + 1 \leqslant \bar{x} \leqslant (a_n + 1)10^n.$$

Această dublă inegalitate poate fi scrisă și astfel :

$$a_n 10^n \leqslant \bar{x} - 1 \text{ și } \bar{x} \leqslant (a_n + 1)10^n.$$

Ținînd seama că

$$\bar{x} - 1 \leqslant x \text{ și } x \leqslant \bar{x}$$

care au fost scrise mai înainte avem

$$a_n 10^n \leqslant \bar{x} - 1 \leqslant x \leqslant \bar{x} \leqslant (a_n + 1)10^n.$$

Reținînd numai relațiile privitoare la numărul real x se găsește

$$a_n 10^n \leqslant x < (a_n + 1)10^n.$$

Pentru aproximanta x^* putem scrie relația direct, deoarece demonstrația pentru $\bar{x}$ este aceeași, oricare ar fi ordinul la care se face rotunjirea prin adaos. Așa încît

$$a_n 10^n \leqslant x^* \leqslant (a_n + 1)10^n.$$

Această a doua dublă relație scrisă pentru x este numită *a doua dublă inegalitate* fundamentală în aproximarea numerelor reale.

De subliniat este faptul că cele două duble inegalități fundamentale în aproximarea numerelor se referă de fapt numai la numărul real x, dar am introdus alături de numărul real x și o aproximantă a lui, x^* care poate avea totdeauna o scriere zecimală finită și în consecință poate substitui în calcule pe x.

Precizăm că toate cele arătate în acest paragraf au fost considerate pentru cazul în care numărul real x este pozitiv. Pentru cazul în care numărul real x ar fi negativ se păstrează toate afirmațiile, schimbînd doar inegalitățile în inegalități de sens contrar.

III.7. Aproximante cu eroarea absolută mai mică sau egală cu $\frac{1}{10^k}$

Fie numărul real x iar x_1^* și x_2^* două aproximante ale lui x, prima prin lipsă și a doua prin exces.

III.7.1. *Definiție.* **Spunem că x_1^* și x_2^* sînt aproximante prin lipsă și respectiv prin exces ale lui x cu eroarea absolută mai mică sau egală cu $\frac{1}{10^k}$ dacă verifică următoarele 3 proprietăți :**

1° Fiecare au cîte k cifre la partea zecimală, adică $x_1^* = \frac{x'}{10^k}$ și $x_2^* = \frac{x''}{10^k}$ unde x' și x'' sînt întregi.

2° Numărul x se află cuprins între valorile aproximantelor respective, adică $x_1^* \leqslant x \leqslant x_2^*$.

3° Diferența lor este $\frac{1}{10^k}$, adică $x_2^* - x_1^* = \frac{1}{10^k}$.

Eroarea absolută este mai mică sau egală cu $\frac{1}{10^k}$ pentru că din definiție avem

$$x_1^* \leqslant x \leqslant x_2^*$$

iar aceasta duce la

$$x - x_1^* \leqslant x_2^* - x_1^* \text{ și } x_2^* - x \leqslant x_2^* - x_1^*.$$

Avînd în vedere proprietatea 3° rezultă

$$x - x_1^* \leqslant \frac{1}{10^k} \text{ și } x_2^* - x \leqslant \frac{1}{10^k}.$$

În proprietatea 1° făcînd scăderea între membrii egalităţilor şi ţinînd seama de proprietatea 3° găsim că

$$x'' = x' + 1$$

deci x' şi x'' sînt doi întregi consecutivi. Acest rezultat corelat cu proprietatea 2° arată că avem

$$\frac{x'}{10^k} \leqslant x \leqslant \frac{x'+1}{10^k}.$$

Această relaţie duce la stabilirea modului de aproximare a unui număr real oarecare x.

III.7.2. *Regulă.* Aproximanta prin lipsă cu mai puţin de $\frac{1}{10^k}$ se obţine reţinînd din numărul x cifrele de la partea întreagă şi primele k cifre ale părţii zecimale, iar aproximanta prin adaos se obţine adăugînd o unitate de ordinul $\frac{1}{10^k}$ la aproximanta prin lipsă.

Proprietăţile 1° şi 3° se verifică imediat. Mai trebuie arătat că $x \leqslant x_2^*$ pentru că $x \geqslant x_1^*$ este de asemenea imediată.

Într-adevăr

$$x = a_n a_{n-1} \ldots a_1 a_0, a_{-1} a_{-2} \ldots a_{-k} + 0{,}0 \ldots 0 a_{-(k+1)} a_{-(k+2)} \ldots = x_1^* +$$

$$+ 0{,}0 \ldots 0 a_{-(k+1)} a_{-(k+2)} \ldots \leqslant x_1^* + 0{,}\underbrace{0 \ldots 0}_{\text{de } k \text{ ori zero}}(9) = x_1^* + 0{,}\underbrace{0 \ldots 0}_{\text{de } k-1 \text{ ori zero}}1 = x_1^* +$$

$$+ \frac{1}{10^k} = x_2^*.$$

Deci

$$x \leqslant x_2^*.$$

La aproximarea prin rotunjire dată la III.3.1 şi III.3.2, aproximantele sînt cu o eroare absolută mai mică sau egală cu $\frac{1}{2 \cdot 10^k}$ ($k \in \mathbf{Z}$, $k < 0$ pentru cifre de la partea întreagă şi $k > 0$ pentru cifre de la partea fracţionară) deoarece avem în vedere cifra care urmează la dreapta cifrei fixate pentru aproximare (dacă este 0, 1, 2, 3, 4 sau dacă este 5, 6, 7, 8, 9).

Cu regula completării se obţin aproximante cu eroare absolută mai mică sau egală cu $\frac{1}{2 \cdot 10^k}$.

De asemenea aproximarea canonică sau prin trunchiere precum şi aproximarea prin rotunjire efectuate la maşinile de calcul, dau erori mai mici sau egale cu $\frac{1}{10^k}$ şi respectiv $\frac{1}{2 \cdot 10^k}$.

Exemple:

1. Să se aproximeze numărul π cu o aproximantă prin lipsă şi o aproximantă prin exces avînd o eroare absolută cel mult egală cu $\frac{1}{10^k}$ unde $k = 1, 2, 3$, iar $\pi = 3{,}14159265\ldots$

$k = 1$	prin lipsă	$\pi = 3{,}1$
	prin exces	$\pi = 3{,}2$
$k = 2$	prin lipsă	$\pi = 3{,}14$
	prin exces	$\pi = 3{,}15$
$k = 3$	prin lipsă	$\pi = 3{,}141$
	prin exces	$\pi = 3{,}142$

2. Să se aproximeze numărul π cu o eroare absolută cel mult egală cu $\frac{1}{2 \cdot 10^k}$, unde $k = 1, 2, 3, 4$.

$$k = 1 \Rightarrow \pi = 3{,}1\ ;\ k = 2 \Rightarrow \pi = 3{,}14\ ;\ k = 3 \Rightarrow \pi = 3{,}142\ ;$$

$$k = 4 \Rightarrow \pi = 3{,}1416.$$

III.8. Cifre sigure şi cifre îndoielnice pentru o aproximantă dată

Fie numărul real dat de

$$x = a_n 10^n + a_{n-1} 10^{n-1} + \ldots + a_{n-k} 10^{n-k} + a_{n-k-1} 10^{n-k-1} + \ldots$$

şi x^* o aproximantă a lui, conţinînd primele k cifre.

$$x^* = a_n 10^n + a_{n-1} 10^{n-1} + \ldots + a_{n-k+1} 10^{n-k+1}.$$

Eroarea absolută a acestei aproximante este dată de

$$a \leqslant (a_{n-k} + 1) 10^{n-k}.$$

Deoarece $a_{n-k} \leqslant 9$ avem

$$A = 10^{n-k+1}.$$

Pentru cazurile în care dorim ca aproximarea să fie mai bună se poate aplica regula de completare pentru determinarea unei aproximante ce are o eroare mai mică sau egală cu $\frac{1}{2 \cdot 10^k}$ pentru numere pozitive.

Din

$$a \leqslant (a_{n-k} + 1) 10^{n-k}$$

deducem următoarele două cazuri aplicînd regula completării:

1°. pentru $a_{n-k} \geqslant 5$, majorăm pe a_{n-k+1} cu o unitate și obținem o aproximantă prin exces a cărei eroare absolută este dată de

$$a = x^* - x = 10^{n-k+1} - (a_{n-k}10^{n-k} + \ldots)$$

unde punînd în locul cifrei a_{n-k} valoarea minimă pe care o poate lua, 5 și neglijînd celelalte cifre ce urmează după a_{n-k} diferența crește devenind

$$A = x^* - x \leqslant 10^{n-k+1} - 5\cdot 10^{n-k} = \frac{1}{2}10^{n-k+1};$$

2°. pentru $a_{n-k} < 5$ considerăm în relația

$$a \leqslant (a_{n-k} + 1)10^{n-k}$$

cifra $a_{n-k} = 4$, valoarea maximă pe care o poate lua și avem o aproximantă prin lipsă astfel încît eroarea absolută

$$a = x - x^* \leqslant 5\cdot 10^{n-k}.$$

Înmulțind și împărțind cu 2 avem

$$A = \frac{1}{2}10^{n-k+1}.$$

În acord cu aceste ultime două cazuri putem da următoarea

III.8.1. *Definiție.* **Spunem că x^* este o aproximantă a numărului real x $x>0$, cu k cifre sigure dacă eroarea absolută maximă este mai mică sau egală cu 10^{n-k+1} fără aplicarea regulii de completare și mai mică sau egală cu $\frac{1}{2}10^{n-k+1}$ cu aplicarea regulii de completare, unde n este caracteristica numărului x.**

Cifrele care urmează cifrelor sigure se numesc cifre nesigure sau cifre îndoielnice.

De aceea dintr-o aproximantă nu vom reține decît cifrele sigure. În calcule însă este bine să se rețină două sau chiar trei cifre în plus peste numărul de cifre sigure și numai la rezultat luăm aproximanta acestuia cu numărul de cifre sigure precizat de eroarea absolută maximă.

Exemple:

1. Să se determine numărul de cifre sigure ale aproximantei numărului π a cărei eroare absolută maximă este de 0,00001.

Avem din problemă:

$$A = \frac{1}{10^5},$$

iar din definiție,

$$A = 10^{n-k+1}$$

fără aplicarea regulii de completare.

Deoarece numărul π are o singură cifră la partea întreagă, urmează că $n = 0$.

Deci,

$$A = \frac{1}{10^5} = 10^{-k+1}.$$

Adică,

$$10^5\cdot 10^{-k+1} = 1 = 10^0,$$

încît

$$10^{-k+6} = 10^0,\ -k + 6 = 0,\ k = 6.$$

Rezultă că aproximanta respectivă a fost luată cu 6 cifre sigure, deci este 3,14159.

2. Să se determine numărul de cifre îndoielnice ale aproximantei numărului $\sqrt{3} \approx 1{,}7320508076425$ a cărei eroare absolută maximă este de $\frac{1}{10^{10}}$.

Avem

$$A = \frac{1}{10^{10}} = 10^{n-k+1}$$

fără aplicarea regulii de completare.

Deoarece numărul $\sqrt{3}$ are o singură cifră la partea întreagă, înseamnă că $n = 0$.

Deci

$$A = \frac{1}{10^{10}} = 10^{-k+1};\ 10^{10}\cdot 10^{-k+1} = 1 = 10^0$$

încît

$$-k + 11 = 0,\ k = 11.$$

Aproximanta lui $\sqrt{3}$ avînd 11 cifre sigure, urmează că ultimele 3 cifre sînt îndoielnice.

III.9. Determinarea erorilor relative maxime ale unei aproximante pe baza cifrelor sigure ale acesteia

Să considerăm numărul real x și x^* o aproximantă a sa cu k cifre sigure. Fie a_n prima cifră a numărului x și a aproximantei x^*, $(a_n \neq 0)$.

Erorile absolute maxime fără regula de completare și cu regula de completare sînt date respectiv de

$$A = 10^{n-k+1} \text{ și } A = \frac{1}{2}10^{n-k+1}.$$

Erorile relative maxime în aceste cazuri vor fi deduse din

$$E = \frac{A}{|x|}$$

unde vom avea în vedere că $x \geqslant a_n10^n$, care fără regula de completare conduce la

$$E \leqslant \frac{10^{n-k+1}}{a_n10^n} = \frac{1}{a_n10^{k-1}} \text{ deci } E \leqslant \frac{1}{a_n10^{k-1}},$$

iar cu regula de completare conduce la

$$E \leqslant \frac{10^{n-k+1}}{2a_n 10^n} = \frac{1}{2a_n 10^{k-1}} \text{ deci } E \leqslant \frac{1}{2a_n 10^{k-1}}.$$

Din cele demonstrate rezultă :

Teoremă. **Fiind dat un număr x şi o aproximantă a sa x^* cu k cifre sigure, eroarea relativă maximă a lui x^* este mai mică sau egală cu $\frac{1}{a_n 10^{k-1}}$ fără regula de completare şi mai mică sau egală cu $\frac{1}{2a_n 10^{k-1}}$ cu regula de completare, unde a_n este prima cifră nenulă de la stînga lui x^*.**

Este uşor de remarcat că erorile relative maxime exprimate în procente sau nu, pot fi antecalculate deoarece depind numai de prima cifră a numărului şi de numărul de cifre sigure, astfel încît se poate întocmi un tabel de erori relative maxime. Astfel de tabele sînt întocmite şi se află în multe formulare matematice de unde ele pot fi luate pentru a fi folosite.

Dăm şi noi un astfel de tabel în care sînt trecute erorile relative maxime în procente.

Tabelul III.1

Erorile relative maxime cînd se cunoaşte a_n şi k cu regula de completare

Prima cifră semnificativă a_n	Numărul de cifre sigure k							
	1	2	3	4	5	6	7	8
1	50	5,0	0,50	0,050	0,0050	0,00050	0,000050	0,0000050
2	25	2,5	0,25	0,025	0,0025	0,00025	0,000025	0,0000025
3	17	1,7	0,17	0,017	0,0017	0,00017	0,000017	0,0000017
4	12	1,2	0,12	0,012	0,0012	0,00012	0,000012	0,0000012
5	10	1,0	0,10	0,010	0,0010	0,00010	0,000010	0,0000010
6	8,3	0,83	0,083	0,0083	0,00083	0,000083	0,0000083	0,00000083
7	7,1	0,71	0,071	0,0071	0,00071	0,000071	0,0000071	0,00000071
8	6,3	0,63	0,063	0,0063	0,00063	0,000063	0,0000063	0,00000063
9	5,6	0,56	0,056	0,0056	0,00056	0,000056	0,0000056	0,00000056

Exemple :
Fie numărul
$x = 27{,}24536$
care are primele 5 cifre sigure. Să se determine eroarea relativă maximă prin calcul şi apoi confruntînd rezultatul cu tabelul de valori.

Din calcul ştim că fără regula de completare avem

$$E = \frac{1}{a_n 10^{k-1}}$$

unde, pentru cazul nostru, $a_n = 2$ şi $k = 5$. Deci,

$$E = \frac{1}{2 \cdot 10^4} = 0{,}00005.$$

Cu regula de completare, avem

$$E = \frac{1}{2a_n 10^{k-1}}$$

care, cu datele problemei, dă

$$E = \frac{1}{4 \cdot 10^4} = 0{,}000025,$$

adică

$$E = 0{,}0025\,\%,$$

căutînd în tabel pe linia 2, coloana 5 găsim acelaşi rezultat.

III.10. Determinarea numărului de cifre sigure ale unei aproximante cînd se cunoaşte eroarea relativă maximă a ei

Să considerăm prin ipoteză, eroarea relativă a aproximantei x^* a lui x cu eroare relativă maximă, egală cu :

$$E = \frac{1}{s \cdot 10^p}, \text{ unde } 1 \leqslant s \leqslant 9.$$

Folosind rezultatele din paragraful precedent deducem că numărul k de cifre sigure ale aproximantei x^* este cel mai mare număr natural k pentru care este verificată inegalitatea :

$$\frac{1}{s \cdot 10^p} \leqslant \frac{1}{a_n 10^{k-1}},$$

adică,

$$s \cdot 10^p \geqslant a_n \cdot 10^{k-1},$$

ceea ce este echivalent cu

$$10^{k-p-1} \leqslant \frac{s}{a_n}.$$

a) Dacă $s \geqslant a_n$,

rezultă

$$10^{k-p-1} \leqslant 1 \leqslant \frac{s}{a_n}.$$

Deci trebuie să avem,

$$k - p - 1 \leqslant 0,$$

de unde deducem $k \leqslant p + 1$.

b) Dacă $s < a_n$,

rezultă

$$10^{k-p-1} \leqslant \frac{1}{10} < \frac{s}{a_n},$$

de unde

$$k - p - 1 \leqslant -1 \text{ sau } k \leqslant p.$$

Avem deci:

Teoremă. **Dacă eroarea relativă maximă a aproximantelor unui număr este** $\frac{1}{s \cdot 10^p}$ **(s şi p date) cu $1 \leqslant s \leqslant 9$ şi $p \in \mathbf{N}$, aproximantele au $p+1$ cifre sigure cînd $s \geqslant a_n$ şi p cifre sigure cînd $s < a_n$.**

Întocmai ca şi în cazul precedent se pot întocmi tabele pentru determinarea cifrelor sigure pentru erori relative maxime date.

Dăm în continuare tabelul cu cifrele sigure pentru cîteva valori ale erorii relative maxime pentru

$$a_n = 1, 2, 3, \ldots, 9.$$

Tabelul III.2

a_n	E			
	5%	1%	0,5%	0,1%
1	2	3	3	4
2	2	2	3	3
3	1	2	2	3
4	1	2	2	3
5	1	2	2	3
6	1	2	2	3
7	1	2	2	3
8	1	2	2	3
9	1	2	2	3

Exemple:

1. Fie $x^* = 24{,}38256$ o aproximantă a unui număr x care are eroarea relativă $E = \frac{1}{50\,000}$. Să se afle numărul de cifre sigure.

$$E = \frac{1}{5 \cdot 10^4}$$

deci $s = 5$ şi $p = 4$. Avînd $a_n = 2 < s$ rezultă $k = p + 1 = 5$, încît x^* are primele 5 cifre din stînga sigure.

2. Fie aproximantele lui π date cu o eroare relativă maximă de 0,5%. Să se afle numărul de cifre sigure ale aproximantelor.

Folosind tabelul III.2 se găseşte la intersecţia liniei a treia cu coloana 0,5% numărul 2, deci aproximanta are două cifre sigure.

III.11. Efectuarea sumelor şi diferenţelor cu aproximante

Aproximantele fiind numere raţionale operaţiile de adunare şi scădere cu acestea nu se deosebesc cu nimic de aceleaşi operaţii efectuate cu numerele raţionale. Ceea ce interesează în mod deosebit este eroarea transmisă la rezultat.

Fie numerele reale x_1 şi x_2 şi aproximantele lor x_1^* şi x_2^* cu erorile absolute a_1 şi a_2. Vom considera ca aproximantă a sumei, suma celor două aproximante şi deci eroarea absolută a sumei va fi dată de

$$|(x_1^* + x_2^*) - (x_1 + x_2)| = a.$$

Această relaţie se mai poate scrie

$$|(x_1^* - x_1) + (x_2^* - x_2)| \leqslant |x_1^* - x_1| + |x_2^* - x_2| = a_1 + a_2,$$

adică,

$$a \leqslant a_1 + a_2.$$

Este uşor de văzut că putem generaliza această relaţie pentru o sumă cu un număr finit de termeni, fapt ce ne permite să enunţăm rezultatul prin următoarea teoremă.

Teoremă. **Eroarea absolută a aproximantei unei sume finite de termeni, numere reale, este mai mică sau cel mult egală cu suma erorilor absolute ale aproximantelor termenilor.**

Considerăm, de asemenea, numerele reale x_1 şi x_2 şi aproximantele lor x_1^* şi x_2^* erorile absolute maxime fiind A_1 şi A_2 iar ca aproximantă a sumei, suma celor două aproximante. Ne întrebăm care este relaţia între eroarea absolută a sumei şi erorile absolute ale aproximantelor termenilor sumei?

Dacă $x_1^* + x_2^* = x^*$ este o aproximantă a sumei $x_1 + x_2 = x$, eroarea absolută a sumei aproximantelor este dată de

$$|x^* - x| = |(x_1^* + x_2^*) - (x_1 + x_2)| = |(x_1^* - x_1) + (x_2^* - x_2)| \leqslant |x_1^* - x_1| + |x_2^* - x_2|,$$

adică

$$a \leqslant a_1 + a_2$$

şi avînd în vedere definiţia dată pentru eroarea absolută maximă la III.4.2 avem $a_1 \leqslant A_1$ şi $a_2 \leqslant A_2$, deci

$$a_1 + a_2 \leqslant A_1 + A_2,$$

adică

$$a \leqslant A_1 + A_2.$$

Deoarece această relație se poate extinde la o sumă cu un număr finit de termeni, putem da următoarea teoremă :

Teoremă. **Eroarea absolută a sumei este mai mică sau cel mult egală cu suma erorilor absolute maxime ale termenilor sumei.**

De asemenea, putem considera ca eroare maximă a sumei însăși suma erorilor absolute maxime, deoarece notînd

$$A_1 + A_2 = A$$

avem

$$a \leqslant A.$$

Cum și această relație se poate extinde la o sumă cu un număr finit de termeni, și utilizînd în acest caz noțiunea de *eroare absolută maximă a sumei*, avem teorema :

Teoremă. **Eroarea absolută maximă a unei sume este egală cu suma erorilor absolute maxime ale termenilor sumei.**

În ceea ce privește eroarea relativă maximă a lui x^* care reprezintă suma celor două aproximante x_1^* și x_2^* ale numerelor reale x_1 și x_2 considerate, ea este dată de

$$E = \frac{A}{|x|} \text{ sau } |x|E = A.$$

Însă

$$A = A_1 + A_2$$

și

$$A_1 = |x_1|E_1, \quad A_2 = |x_2|E_2,$$

deci

$$|x|E = |x_1|E_1 + |x_2|E_2,$$

sau

$$E = \frac{|x_1|}{|x|} E_1 + \frac{|x_2|}{|x|} E_2,$$

adică

$$E = \frac{|x_1|E_1 + |x_2|E_2}{|x_1 + x_2|}, \text{ deoarece } x = x_1 + x_2.$$

Această relație poate fi generalizată pentru un număr n finit de termeni și utilizînd semnul $\sum$ pentru sumă avem

$$E = \frac{\sum_{i=1}^{n} A_i}{\left| \sum_{i=1}^{n} x_i \right|}.$$

Dacă în exprimarea anterioară a lui E dată pentru două numere presupunem că avem :

$$E_1 \leqslant E_2,$$

atunci

$$\frac{|x_1| + |x_2|}{|x_1 + x_2|} E_1 \leqslant E \leqslant \frac{|x_1| + |x_2|}{|x_1 + x_2|} E_2.$$

Și această relație poate fi generalizată pentru un număr n finit de termeni obținîndu-se

$$\frac{\sum_{i=1}^{n} |x_i|}{\left| \sum_{i=1}^{n} x_i \right|} \cdot \min(E_i) \leqslant E \leqslant \frac{\sum_{i=1}^{n} |x_i|}{\left| \sum_{i=1}^{n} x_i \right|} \max(E_i),$$

unde am notat cu $\max(E_i)$ cea mai mare dintre erorile relative maxime ale termenilor, iar cu $\min(E_i)$ cea mai mică dintre erorile relative maxime ale termenilor sumei.

Ultimele două relații dau posibilitatea evidențierii a două importante observații cu privire la aprecierea erorii relative a aproximantei unei sume.

Observația 1. Eroarea relativă maximă a unei sume de aproximante de același semn este cuprinsă între cea mai mică și cea mai mare dintre erorile relative maxime ale termenilor sumei.

Într-adevăr, pentru toți x_i de același semn avem

$$\sum_{i=1}^{n} |x_i| = \left| \sum_{i=1}^{n} x_i \right|$$

și deci ultima relație devine

$$\min(E_i) \leqslant E \leqslant \max(E_i).$$

Observația 2. Eroarea relativă maximă a unei sume algebrice de aproximante, poate depăși, în general, pe cea mai mare dintre erorile relative maxime ale termenilor sumei.

Într-adevăr, pentru cazul că nu toți x_i au același semn avem

$$\sum_{i=1}^{n} |x_i| > \left| \sum_{i=1}^{n} x_i \right|$$

și deci

$$\frac{\sum_{i=1}^{n} |x_i|}{\left| \sum_{i=1}^{n} x_i \right|} > 1$$

încît există în general posibilitatea ca E să depășească pe $\max(E_i)$.

Exemple :

1. Să se calculeze cu șase cifre sigure suma

$$x = 234{,}172 + 26{,}5736 + 8{,}75735.$$

Pentru fiecare număr, cele șase cifre sînt sigure deci suma poate să aibă cel mult șase cifre sigure așa încît reținem de la cel mai mare număr toate cifrele, iar la celelalte numere aplicăm regula de rotunjire (regula completării) astfel încît fiecare număr să aibă după virgulă atîtea zecimale, cît numărul care are cele mai puține zecimale, astfel,

$$\begin{array}{r} 234{,}172 \\ 26{,}574 \\ 8{,}757 \\ \hline 269{,}503 \end{array}$$

2. Să se calculeze eroarea absolută maximă a sumei

$$x = 4\,597{,}273 + 872{,}9736 + 0{,}47983$$

unde erorile absolute sînt indicate pentru fiecare număr, de ordinul ultimei sale cifre.

Avem deci erorile absolute ale numerelor date de

$$\frac{1}{10^3}, \frac{1}{10^4}, \frac{1}{10^5},$$

așa încît eroarea absolută maximă este

$$\frac{1}{10^3} + \frac{1}{10^4} + \frac{1}{10^5} = \frac{100 + 10 + 1}{10^5} < \frac{200}{10^5} = \frac{2}{10^3}.$$

Aplicînd regula completării, avem,

$$\begin{array}{r} 4\,597{,}273 \\ 872{,}974 \\ 0{,}480 \\ \hline 5\,470{,}727 \end{array}$$

și eroarea absolută este mai mică decît $\frac{2}{10^3}$.

De remarcat că și în cazul aproximării prin trunchiere (aproximarea canonică) eroarea absolută a sumei rămîne mai mică decît eroarea absolută maximă.

La aceeași aproximare a sumei ajungem și dacă considerăm la termenii sumei o zecimală peste ordinul cifrei la care se face aproximarea, adică în loc ca ultima cifră să fie cifra care dă eroarea absolută $\frac{1}{10^n}$, luăm ca ultimă cifră zecimală pe aceea care dă eroarea $\frac{1}{10^{n+1}}$, iar după efectuarea sumei aplicăm rezultatului metoda completării.

Deoarece eroarea absolută maximă a unei sume aritmetice este egală cu suma erorilor absolute maxime ale termenilor, erorile foarte mici ale unor termeni pot fi neglijate. De aceea nu are sens să păstrăm prea multe cifre la termenii cu o precizie mai mare.

Să analizăm acum cazul diferenței a două numere reale. Fie x_1^* și x_2^* aproximantele numerelor reale x_1 și x_2, iar a_1 și a_2 erorile absolute ale aproximantelor. Deoarece diferența $x = x_1 - x_2$ și o aproximantă a sa $x^* = x_1^* - x_2^*$ sînt sume algebrice, unele rezultate obținute pentru suma algebrică sînt aplicabile și în cazul diferenței. Utilizînd noțiunea de *eroare absolută maximă a unei diferențe* avem următoarea teoremă pe care o enunțăm fără a o demonstra :

Teoremă. **Eroarea absolută maximă a unei diferențe este egală cu suma erorilor absolute maxime ale termenilor diferenței.**

Tot din rezultatele obținute la suma algebrică vom remarca distingerea a două cazuri care rezultă din *Observația 2*, făcută în acest paragraf :

a) cînd descăzutul și scăzătorul, avînd același semn, diferă mult ca mărime a valorilor absolute, atunci eroarea relativă maximă a diferenței se determină întocmai ca pentru sumă ;

b) cînd descăzutul și scăzătorul, avînd același semn, au valorile absolute foarte apropiate, erori mici ale termenilor pot atrage după ele o eroare relativă maximă foarte mare a diferenței.

Exemple :

1. Să se calculeze diferența

$$x = x_1 - x_2,$$

unde

$$x_1 = 18{,}2379 \text{ și } x_2 = 18{,}2374$$

sînt numere cu valori apropiate.

Considerînd aproximantele obținute prin rotunjire la cifra miimilor avem

$$x_1^* = 18{,}238,\ x_2^* = 18{,}237$$

cu erorile absolute

$$a_1 = 0{,}0001 \text{ și } a_2 = 0{,}0004.$$

Rezultă erorile relative ale termenilor

$$e_1 = \frac{0{,}0001}{18{,}2379} = 0{,}000005\ ;\ e_2 = \frac{0{,}0004}{18{,}2374} = 0{,}00002.$$

Dar

$$x = x_1 - x_2 = 0{,}0005 \text{ și } x^* = x_1^* - x_2^* = 0{,}001.$$

Deoarece

$$a = |x^* - x| \text{ și } e = \frac{a}{|x|},$$

avem

$$a = 0{,}0005$$

și eroarea relativă a diferenței

$$e = \frac{0{,}0005}{0{,}0005} = 1$$

care arată că eroarea relativă a diferenței este foarte mare în comparație cu erorile relative ale termenilor.

2. Să se calculeze diferența

$$x = x_1 - x_2,$$

unde

$$x_1 = 23{,}4759,\ x_2 = 3{,}4753$$

sînt numere cu valori absolute ce diferă mult între ele.

Considerînd aproximantele obținute prin rotunjire la cifra miimilor avem

$$x_1^* = 23{,}476 \text{ și } x_2^* = 3{,}475$$

cu erorile absolute

$$a_1 = 0{,}0001 \text{ și } a_2 = 0{,}0003,$$

rezultă

$$e_1 = \frac{0{,}0001}{23{,}4759} = 0{,}000004 \text{ și } e_2 = \frac{0{,}0003}{3{,}4753} = 0{,}00008$$

dar

$$x = x_1 - x_2 = 20{,}0006 \text{ și } x^* = x_1^* - x_2^* = 20{,}001$$

deci

$$a = 0{,}0004,$$

de unde

$$e = \frac{0{,}0004}{20{,}0006} = 0{,}00002$$

de unde se vede că eroarea relativă este foarte apropiată de erorile relative ale termenilor.

Presupunînd, deci, că cei doi termeni ai diferenței, avînd același semn, au diferența mare între module, ținîndu-se seama de rezultatele obținute la sumă dăm următoarea regulă fără demonstrație :

III.11.1. — *Regulă.* Pentru calcularea diferenței între două aproximante, care aproximează două numere cu erori absolute maxime $\frac{1}{10^n}$ și respectiv $\frac{1}{10^{n+m}}$ ($n, m \in \mathbf{N}$), se reține cîte n zecimale de la fiecare aproximantă (prin lipsă sau adaos) și se face diferența. Eroarea absolută maximă a diferenței este de $\frac{1}{10^n}$.

Exemple :

1. Să se calculeze diferența

$$0{,}035746 - 0{,}02379$$

știind că erorile absolute maxime sînt de $\frac{1}{10^4}$ și respectiv $\frac{1}{10^3}$. Reținînd cîte trei zecimale la ambele numere avem aproximantele cu care efectuăm diferența. Eroarea absolută maximă a diferenței este $\frac{1}{10^3}$

$$0{,}036 - 0{,}024 = 0{,}012.$$

2. Să se calculeze diferența

$$543{,}245 - 0{,}214312$$

știind că erorile absolute maxime sînt date de $\frac{1}{10^3}$ și respectiv $\frac{1}{10^6}$. Reținem deci cîte trei zecimale la fiecare și cu aproximantele respective efectuăm diferența. Eroarea absolută maximă a diferenței este $\frac{1}{10^3}$

$$543{,}245 - 0{,}214 = 543{,}031.$$

3. Să se calculeze eroarea relativă a diferenței

$$573{,}23746 - 573{,}23742$$

știind că erorile absolute sînt, la ambele numere, mai mici decît $0{,}5 \cdot \frac{1}{10^5}$.

Deoarece diferența lor este foarte mică, 0,00004, iar eroarea absolută a diferenței fiind mai mică decît $0{,}5\frac{1}{10^5}$, nu putem estima eroarea relativă a rezultatului (observația 2 din acest paragraf).

III.12. Efectuarea produselor și citurilor cu aproximante

Să considerăm numerele reale x_1 și x_2 cu aproximantele x_1^* și x_2^* avînd erorile absolute a_1 și, respectiv, a_2. Fie $x^* = x_1^* \cdot x_2^*$ o aproximantă a lui $x = x_1 \cdot x_2$ cu eroare absolută

$$a = |\, x_1^* \cdot x_2^* - x_1 \cdot x_2 \,|$$

avem

$$x_1 = x_1^* \pm a_1 \text{ și } x_2 = x_2^* \pm a_2,$$

deci

$$x_1 \cdot x_2 = x_1^* \cdot x_2^* \pm x_1^* \cdot a_2 \pm x_2^* \cdot a_1 \pm a_1 \cdot a_2.$$

Calculînd $x_1^* \cdot x_2^* - x_1 \cdot x_2$ și trecînd la valori absolute avem $a = |\, x_1^* \cdot x_2^* - x_1 \cdot x_2 \,| = |\, \pm x_1^* \cdot a_2 \pm x_2^* \cdot a_1 \pm a_1 \cdot a_2 \,|$,

de unde

$$a \leqslant |\, x_1^* \,| a_2 + |\, x_2^* \,| a_1 + a_1 a_2,$$

care arată că eroarea absolută a produsului de doi factori este mai mică sau cel mult egală cu suma produselor dintre un factor și eroarea absolută a celuilalt mărită cu produsul erorilor absolute ale celor doi factori.

Împărțind relația precedentă cu $|\, x \,| = |\, x_1 \,| \cdot |\, x_2 \,|$ avem

$$\frac{a}{|\, x \,|} \leqslant \frac{|\, x_1^* \,| \cdot a_2}{|\, x_1 \,| \cdot |\, x_2 \,|} + \frac{|\, x_2^* \,| \cdot a_1}{|\, x_1 \,| \cdot |\, x_2 \,|} + \frac{a_1 \cdot a_2}{|\, x_1 \,| \cdot |\, x_2 \,|}.$$

Ţinînd seama de faptul că $\frac{|x_1^*|}{|x_1|}$ şi $\frac{|x_2^*|}{|x_2|}$ au valorile foarte apropiate de valoarea 1, precum şi că produsul $\frac{a_1}{|x_1|}\cdot\frac{a_2}{|x_2|}$ fiind foarte mic, în comparaţie cu suma lor, poate fi neglijat, se poate scrie între erorile relative maxime relaţia

$E \approx E_1 + E_2$, unde semnul „$\approx$" înseamnă „aproximativ egal". Utilizînd noţiunea de *eroare relativă maximă a produsului* în cazul produsului cu aproximante avem teorema :

Teoremă. **Eroarea relativă maximă a produsului a două aproximante este aproximativ egală cu suma erorilor relative maxime ale aproximantelor.**

Generalizarea acestui rezultat se poate uşor extinde la orice produs cu un număr finit de factori şi în acest caz eroarea relativă maximă a produsului este aproximativ egală cu suma erorilor relative maxime ale factorilor.

În cazul în care erorile relative pentru n factori sînt valori apropiate, eroarea relativă maximă a produsului va fi aproximativ egală cu de n ori cea mai mare dintre erorile relative ale factorilor. Invers, ca să obţinem produsul, cu o anumită eroare relativă, a n factori avînd valori apropiate, trebuie să luăm factorii cu erori relative de n ori mai mici decît eroarea relativă a produsului.

Propoziţie : Produsul are cu una sau cel mult două cifre sigure mai puţin decît cel mai mic dintre numerele de cifre sigure ale tuturor factorilor şi invers pentru ca produsul să aibă n cifre sigure trebuie să luăm toţi factorii cu $n+1$ sau $n+2$ cifre sigure.

Demonstraţie :

Fie, de exemplu, x_1^* şi x_2^* două aproximante cu n cifre sigure şi se cere să se calculeze numărul de cifre sigure ale produsului lor.

Dacă $x^* = x_1^* \cdot x_2^*$, atunci $E \approx E_1 + E_2$,
adică

$$\frac{1}{10^{n-1}} + \frac{1}{10^{n-1}} = \frac{2}{10^{n-1}} = \frac{10}{5\cdot 10^{n-1}} < \frac{1}{10^{n-2}}$$

deci în cel mai nefavorabil caz produsul va avea $n-2$ cifre sigure.

Exemple :

1. Să se calculeze produsul

$$x = 1{,}2478 \cdot 0{,}9735 \cdot 2{,}8679$$

cu o eroare relativă maximă de $\frac{1}{10^2}$, adică cu două zecimale sigure.

Luînd mai întîi factorii cu toate zecimalele avem :

$$x = 1{,}2478 \times 0{,}9735 \times 2{,}8679 = 3{,}483733631070$$

la care reţinem doar primele două zecimale aplicînd rotunjirea, deci,

$$x^* = 3{,}48.$$

Considerînd numai cîte trei zecimale la fiecare număr (adică o zecimală în plus faţă de precizia cerută pentru produs) avem :

$$x^* = 1{,}247 \times 0{,}973 \times 2{,}867 = 3{,}478619977,$$

unde, reţinînd numai două zecimale aplicînd rotunjirea, avem

$$x^* = 3{,}48$$

în care se poate vedea că primele două cifre zecimale coincid cu ale rezultatului corect.

Considerînd însă numai două cifre zecimale la fiecare număr avem

$$x = 1{,}24 \times 0{,}97 \times 2{,}86 = 3{,}440008$$

din care se vede că a doua zecimală este eronată.

2. Să se determine produsul şi eroarea relativă maximă a produsului

$$x = 1{,}4750 \cdot 1{,}2864$$

ştiind că factorii au eroarea relativă maximă de $\frac{1}{10^4}$ şi $\frac{1}{10^4}$, vom considera

$$x = 1{,}475 \cdot 1{,}286 = 1{,}89685,$$

unde vom reţine numai două zecimale după aplicarea rotunjirii deci
$x^* = 1{,}90$ cu eroarea relativă maximă de $\frac{1}{10^2}$.

Să cercetăm acum cazul cîtului de două numere reale.

Fie x_1 şi x_2, cele două numere, x_1^* şi x_2^* aproximantele lor cu erorile absolute a_1 şi a_2. Notăm cu $x^* = \frac{x_1^*}{x_2^*}$ o aproximantă a lui $x = \frac{x_1}{x_2}$ cu eroarea absolută a.

Avem

$$x - x^* = \frac{x_1}{x_2} - \frac{x_1^*}{x_2^*} = \frac{x_1^* \pm a_1}{x_2^* \pm a_2} - \frac{x_1^*}{x_2^*} = \frac{x_1^* \cdot x_2^* \pm x_2^* \cdot a_1 - x_1^* \cdot x_2^* \pm x_1^* a_2}{x_2^*(x_2^* \pm a_2)},$$

adică

$$a = \frac{|\pm x_1^* \cdot a_2 \pm x_2^* \cdot a_1|}{|x_2^*(x_2^* \pm a_2)|}$$

sau

$$a \leqslant \frac{|x_1^*| \cdot a_2 + |x_2^*| \cdot a_1}{|x_2^*| \cdot |x_2^* \pm a_2|} = \frac{|x_1^*| \cdot a_2 + |x_2^*| \cdot a_1}{|x_2^*| \cdot |x_2|}.$$

Împărţind ultima relaţie cu $\frac{|x_1|}{|x_2|}$ avem

$$e \leqslant \frac{|x_2| \cdot |x_1^*| \cdot a_2}{|x_1| \cdot |x_2| \cdot |x_2^*|} + \frac{|x_2| \cdot |x_2^*| \cdot a_1}{|x_1| \cdot |x_2| \cdot |x_2^*|}.$$

Prin simplificare se obține

$$e \leqslant \frac{|x_1^*|}{|x_1|} \cdot \frac{a_2}{|x_2^*|} + \frac{a_1}{|x_1|}.$$

Însă cum

$$\frac{|x_1^*|}{|x_1|} \approx 1 \text{ și } \frac{a_2}{|x_2^*|} \approx \frac{a_2}{|x_2|},$$

obținem

$$e \approx \frac{a_2}{|x_2|} + \frac{a_1}{|x_1|},$$

unde considerînd erori relative maximale avem

$$E \approx E_1 + E_2.$$

Avînd în vedere ultima relație obținută se poate afirma că eroarea relativă maximă a cîtului de două aproximante este aproximativ egală cu suma erorilor relative maxime ale aproximantelor.

Propoziție : Rezultatul împărțirii a două aproximante are cu una sau cel mult două cifre sigure mai puțin decît cel mai mic dintre numerele de cifre sigure ale deîmpărțitului și împărțitorului și invers, pentru ca la cît să obținem n cifre sigure trebuie să luăm numerele cu $n + 1$ sau $n + 2$ cifre sigure.

Demonstrație :

Fie aproximantele x_1^* și x_2^* avînd n cifre sigure. Să se calculeze numărul de cifre sigure ale cîtului lor.

Dacă $x^* = \frac{x_1^*}{x_2^*}$, atunci $E \approx E_1 + E_2$,

adică

$$\frac{1}{10^{n-1}} + \frac{1}{10^{n-1}} = \frac{2}{10^{n-1}} = \frac{10}{5 \cdot 10^{n-1}} < \frac{1}{10^{n-2}},$$

deci în cazul cel mai nefavorabil cîtul va avea $n - 2$ cifre sigure.

Exemple :

1. Să se calculeze cîtul

$$x = \frac{1{,}26472}{0{,}25796}$$

cu o eroare relativă maximă de $\frac{1}{10^3}$, adică cu două zecimale sigure.

Efectuînd mai întîi cîtul cu numerele nerotunjite avem

$$x = 1{,}26472 : 0{,}25796 = 4{,}90277...$$

reținînd numai primele două zecimale aplicînd rotunjirea se obține

$$x^* = 4{,}90.$$

Considerînd aproximantele deîmpărțitului și împărțitorului cu cîte 4 zecimale vom avea

$$x^* = 1{,}2647 : 0{,}2580 = 4{,}900,$$

unde se vede că avem cîtul cu primele zecimale același ca mai înainte.

Efectuînd împărțirea între două aproximante avînd trei zecimale se obține

$$x^* = 1{,}265 : 0{,}258 = 4{,}903,$$

unde primele două zecimale sînt aceleași ca și în cazurile de mai înainte.

Considerînd însă aproximantele numai cu primele două zecimale avem

$$x^* = 1{,}26 : 0{,}25 = 5{,}04,$$

ceea ce arată că rezultatul este eronat.

2. Să se calculeze cîtul

$$x = \frac{2{,}43569432}{1{,}264572973}$$

cu o eroare relativă a cîtului de $\frac{1}{10}$. În acest caz aproximantele deîmpărțitului și împărțitorului nu vor avea mai mult de trei zecimale adică o eroare absolută $\frac{1}{10^3}$.

Deci

$$x_1^* = 2{,}436 \text{ și } x_2^* = 1{,}265$$

și avem

$$\frac{x_1^*}{x_2^*} = \frac{2{,}436}{1{,}265} = 1{,}92.$$

Rezultă : cîtul cu eroarea absolută de $\frac{1}{10}$ este dat de

$$x^* = \frac{x_1^*}{x_2^*} = 1{,}9.$$

III.13. Ridicarea la putere a aproximantelor

Pentru ridicarea la o putere, de exponent număr natural n, a aproximantei unui număr real x, putem să considerăm că avem produsul cu n factori egali între ei.

De aceea, fie x numărul real și x^* o aproximantă a sa care are eroarea absolută a. Este normal să considerăm pentru x^n aproximanta $(x^*)^n$ astfel încît avem

$$x^n = (x^* \pm a)^n.$$

Pentru a intui rezultatul într-o astfel de operație vom analiza cazurile :

1. Cînd exponentul $n = 2$, avem

$$x^2 = (x^* \pm a)^2 = (x^*)^2 \pm 2x^* \cdot a + a^2,$$

de unde

$$x^2 - (x^*)^2 = \pm 2x^* \cdot a + a^2.$$

Notînd cu $a^{(2)}$ eroarea absolută a puterii de exponent 2, rezultă

$$a^{(2)} = |x^2 - (x^*)^2| = |\pm 2x^* \cdot a + a^2|,$$

şi avînd în vedere că a^2 este o cantitate ce poate fi neglijată comparînd-o cu a, avem :

$$a^{(2)} \leqslant 2|x^*|a,$$

care împărţită cu $|x|^2$ devine

$$\frac{a^{(2)}}{|x|^2} \leqslant 2\frac{|x^*|}{|x|} \cdot \frac{a}{|x|}.$$

Notînd cu $e^{(2)}$ eroarea relativă a puterii de exponent 2 şi avînd în vedere că $\frac{|x^*|}{|x|} \approx 1$ avem :

$$e^{(2)} \approx 2e \text{ unde } e = \frac{a}{|x|}.$$

Avem deci :

Eroarea relativă a puterii de exponent 2 este aproximativ egală cu dublul erorii relative a bazei.

2. Cînd exponentul $n = 3$, avem

$$x^3 = (x^* \pm a)^3 = (x^*)^3 \pm 3(x^*)^2 a + 3x^* a^2 \pm a^3$$

sau

$$|x^3 - (x^*)^3| = |\pm 3(x^*)^2 \cdot a + 3x^* a^2 \pm a^3|.$$

Notînd cu $a^{(3)}$ eroarea absolută a puterii de exponent 3, rezultă

$$a^{(3)} \leqslant 3|x^*|^2 a + 3|x^*|a^2 + a^3.$$

Împărţind cu $|x|^3$, se obţine

$$\frac{a^{(3)}}{|x|^3} \leqslant 3\frac{|x^*|^2}{|x|^2} \cdot \frac{a}{|x|} + 3\frac{|x^*|}{|x|} \cdot \frac{a^2}{|x|^2} + \frac{a^3}{|x|^3}.$$

Deoarece $\frac{a^{(3)}}{|x|^3}$ reprezintă eroarea relativă, pe care o notăm cu $e^{(3)}$ şi ţinînd seama de valorile neînsemnate pe care le au termenii $\frac{a^2}{|x|^2}$ şi $\frac{a^3}{|x|^3}$ precum şi de $\frac{|x^*|}{|x|} \approx 1$, avem

$$e^{(3)} \approx 3e.$$

De unde, rezultă că :

Eroarea relativă a puterii de exponent 3 este aproximativ egală cu de 3 ori eroarea relativă a bazei.

Este evident că relaţiile au loc şi în cazul erorilor relative maxime care sînt nişte majorante ale erorilor relative.

Adică

$$E^{(2)} \approx 2E \text{ şi } E^{(3)} \approx 3E.$$

Avînd în vedere că puterea de exponent n, număr natural, a unui număr oarecare este produsul de n factori egali cu numărul dat, putem extinde regula produsului de n factori la puterea de exponent n.

Deci putem considera

$$E^{(3)} \approx E^{(2)} + E \approx 2E + E = 3E,$$

$$E^{(4)} \approx E^{(3)} + E \approx 3E + E = 4E,$$

. . . . . .

$$E^{(n)} \approx E^{(n-1)} + E \approx (n-1)E + E = nE.$$

Pentru aplicaţii în cazul puterilor de exponent 2 sau 3 se pot folosi rezultatele obţinute la înmulţire şi de aceea nu vom indica alte procedee.

EXERCIŢII ŞI PROBLEME

1. Să se scrie cîte două aproximante pentru numerele :

8 752,9746 ; 0,0007578945.

2. Ce fel de aproximantă este zero pentru numărul real x ?

3. Dîndu-se două numere raţionale x_1 şi x_2 astfel încît $x_1 < x_2$, care dintre ele poate fi considerat aproximantă pentru celălalt ?

4. Se dă numărul $x = \overline{2,7abc}$ şi o aproximantă a sa $x^* = 2,797$. Ce cifre pot fi a, b, c pentru ca x^* să fie :

1°. o aproximantă prin lipsă ;

2°. o aproximantă prin exces.

5. Să se aproximeze prin rotunjire numerele :

478,956025 ; 1,0047075 ; −7834500 ; −2,653049,

păstrîndu-se primele k zecimale şi să se spună tipul aproximării (lipsă sau exces) unde $k = 1, 2, 3, 4, 5$.

6. Să se aproximeze prin rotunjire numerele :

754 795,26 ; −465 376,572,

la cifra a n-a de la virgulă la stînga pentru $n = 1, 2, 3, 4, 5$, precizîndu-se tipul rotunjirii.

7. Să se facă aproximarea prin trunchiere şi să se precizeze felul aproximării pentru numerele :

0,001754 ; 45,275476 ; −175,4827 ; −0,020765,

păstrîndu-se k zecimale unde $k = 1, 2, 3, 4, 5, 6$.

8. Să se facă aproximarea canonică (trunchiere) a numerelor :

1 794 536 ; −97 453 894 ; 6 285 439,799 ; −14 579 372,4568,

la cifra a n-a de la virgulă spre stînga pentru $n = 1, \ldots, 6$ și să se precizeze de fiecare dată ce fel de aproximante sînt.

9. Dacă $x = \overline{a,bcd}$ și aproximanta sa $x^* = \overline{8,2b}$ care pot fi cifrele a, b, c, d pentru ca :

1°. x^* să fie o aproximantă prin lipsă ;

2°. x^* să fie o aproximantă prin exces ?

10. Se dau numerele $x_1 = \overline{3a4,5}$ și $x_2 = \overline{3ba}$. Să se determine cifrele a și b astfel ca x_1 să fie aproximantă pentru x_2 :

1°. prin lipsă ;

2°. prin exces.

Poate fi x_2 o aproximantă pentru x_1 prin lipsă sau prin exces ?

11. Se dau numerele $x_1 = \overline{3,a5}$ și $x_2 = \overline{3,5a}$. Care sînt cifrele ce pot fi puse în locul literei a pentru ca :

1°. x_1 să fie o aproximantă prin lipsă pentru x_2 ;

2°. x_2 să fie o aproximantă prin lipsă pentru x_1 ?

12. Se dau numerele $x_1 = \overline{0,0\ldots0aabb}$ și $x_2 = \overline{0,0\ldots0abab}$. Care este
k de zero k de zero

relația între a și b pentru ca :

1°. x_1 să fie o aproximantă prin lipsă pentru x_2 ;

2°. x_1 să fie o aproximantă prin exces pentru x_2 ?

13. Se dau numerele :

$x_1 = \overline{0,0\ldots0abc}$;

$x_2 = \overline{0,0\ldots0acb}$;

$x_3 = \overline{0,0\ldots0bac}$; (Fiecare număr are k zerouri consecutive

$x_4 = \overline{0,0\ldots0bca}$; după virgulă.)

$x_5 = \overline{0,0\ldots0cab}$;

$x_6 = \overline{0,0\ldots0cba}$.

Știind că x_1 este o aproximantă prin lipsă pentru x_2 și x_3 o aproximantă prin exces pentru x_5, să se cerceteze ce fel de aproximante formează fiecare număr în parte pentru celelalte ?

14. Știind că $x_1 = \overline{0,5731a}$ este o aproximantă prin rotunjire pentru $x_2 = \overline{0,5731b7}$ și că a este un pătrat perfect, să se stabilească ce cifre pot fi literele a, b ? Folosind configurația grafică a punctelor $P(a, b)$ unde a și b sînt cifrele stabilite mai înainte, să se arate că acestea sînt coliniare.

15. Viteza luminii în vid este $c = 299{,}776 \cdot 10^6$ m/s. În calcul este considerată 300 000 km/s. Care sînt erorile absolută și relativă ale aproximantei ?

16. Să se determine aproximantele prin lipsă, respectiv prin adaos, cu o eroare absolută mai mică decît 0,0001 ale numerelor :

a) 543,054965 ; b) 4,29208759 ; c) −65,493455 ; d) 59,7829736 ;

e) 5 792 736,27 ; f) −0,003.

17. Să se scrie cele două inegalități fundamentale în aproximarea numerelor precum și caracteristica pentru următoarele numere :

a) 4 753,5736 ; b) 1,756304 ; c) 0,5739843 ; d) 0,0004869 ;

e) −18 763,965 ; f) −7,29463 ; g) −0,826907 ; h) −0,00001.

18. Să se calculeze aproximanta lui $\sqrt{2}$ și $\sqrt{3}$ cu eroarea absolută mai mică decît $\frac{1}{2 \cdot 10^3}$.

19. Să se calculeze aproximanta lui $\sqrt{20}$ cu eroarea absolută mai mică decît $\frac{1}{10^3}$.

20. Să se calculeze aproximanta lui π cu eroarea absolută mai mică decît $\frac{1}{2 \cdot 10^6}$.

21. Să se calculeze aproximanta numărului $\frac{22}{7}$ cu eroarea absolută mai mică decît $\frac{1}{10^3}$.

22. Să se calculeze aproximanta numărului $\frac{15}{7}$ cu eroarea absolută mai mică decît $\frac{2}{113}$.

23. Poate fi aproximantă prin lipsă numărul π pentru numărul $\sqrt{10}$ cu o eroare absolută de $\frac{1}{10^2}$?

24. Luînd pentru numerele e și π respectiv aproximantele $e^* = 2{,}71828$ și $\pi^* = 3{,}14159$, să se calculeze erorile lor absolute și relative.

25. Distanța de 37,25 km s-a măsurat cu o precizie de 1 m iar distanța de 78 m s-a măsurat cu o precizie de 1 mm. În ce caz măsurarea a fost mai bună ?

26. Din două cîntăriri succesive apreciem că masa unui corp este de 2,258 kg, iar eroarea absolută maximă este de 1 mg. Care este eroarea relativă maximă ?

27. Măsurînd unghiurile ascuțite ale unui echer găsim mărimile 29°58′ și 60°2′. Știind că mărimile adevărate ale unghiurilor sînt 30° și respectiv 60°, să se calculeze eroarea absolută maximă și eroarea relativă maximă în ambele cazuri.

28. Să se determine eroarea relativă maximă pentru aproximantele numărului π care are k cifre sigure ($k = 1, 2, 3, 4$) fără regula de completare și cu regula de completare folosind tabelul III.1.

29. Se dă numărul $c^* = 2{,}718281828$ o aproximantă a numărului e cu 10 cifre sigure. Să se determine erorile absolute maxime și relative maxime fără regula de completare și cu regula de completare.

30. Se consideră $\frac{7}{11} \approx 0{,}637$. Să se calculeze erorile comise.

31. O aproximantă a lui $\sqrt{83}$ este 9,11. Să se stabilească ce erori s-au comis.

32. Numărul 9,805 este o aproximantă cu 4 cifre sigure a accelerației gravitaționale. Să se determine eroarea relativă maximă a aproximantei.

33. Eroarea absolută maximă a aproximantei lui $\sqrt{13}$ este de 1%. Să se determine numărul de cifre sigure ale acestei aproximante.

34. Să se determine numărul de cifre sigure ale aproximantei lui $\sqrt{1\,491}$, considerînd succesiv că eroarea absolută maximă este 0,1 ; 0,01 ; 0,001 ; 0,0001 și 0,00001, folosind tabelul III.2.

35. Accelerația gravitațională la noi în țară (București) este $g_B = 9{,}805$ m/s². În calcule se utilizează $g = 9{,}8$ m/s². Care este eroarea ? Dar în cazul cînd utilizăm $g = 10$ m/s² ? Care sînt erorile absolută maximă și relativă maximă pentru aceste două aproximante ?

36. Masa electronului în stare de repaus este $m_e = \frac{9{,}108}{10^{28}}$ g. Considerîndu-se în calcule $m_e = \frac{1}{10^{27}}$ g, care este eroarea aproximantei ?

37. Constanta lui Planck este $h = \frac{6{,}625}{10^{34}}$ J·s. Determinați erorile absolute, absolute maxime, relative și relative maxime cînd considerăm aproximantele $h_1 = \frac{7}{10^{34}}$ J·s și $h_2 = \frac{6{,}6}{10^{34}}$ J·s.

38. Valoarea magnetonului Procopiu-Bohr este $\mu_B = \frac{9\,268}{10^{27}} \cdot \frac{J}{T}$. Determinați erorile : absolute, absolute maxime, relative și relative maxime, cînd ca aproximante considerăm $\mu_{B_1} = \frac{927}{10^{26}} \cdot \frac{J}{T}$, $\mu_{B_2} = \frac{93}{10^{25}} \cdot \frac{J}{T}$ și $\mu_{B_3} = \frac{1}{10^{23}} \cdot \frac{J}{T}$.

39. Masa moleculară relativă a acidului etanoic (CH_3COOH) este 60,05. Determinați erorile : absolută, absolută maximă, relativă și relativă maximă, cînd considerăm drept aproximante 60,1 și 60.

40. Determinări experimentale ale greutății specifice a apei distilate (H_2O) au condus la următoarele rezultate ; $\gamma_1 = 9\,808{,}44021$ N/m³, $\gamma_2 = 9\,809{,}7253$ N/m³ și $\gamma_3 = 9\,792{,}342$ N/m³. În rezolvarea problemelor de

chimie sau fizică, descori, considerăm greutatea specifică a apei $\gamma = 9\,810$ N/m³. Să se determine erorile : absolute, absolute maxime, relative și relative maxime ale aproximantei γ, față de valorile indicate prin măsurători.

41. Măsurători ale densității mercurului la presiunea de 760 torri au condus la valorile : $\rho_1 = 13\,595{,}1$ kg/m³ și $\rho_2 = 13\,545{,}7$ kg/m³. În rezolvarea unor probleme, considerăm densitatea mercurului $\rho = 13\,600$ kg/m³. Determinați erorile : absolută, absolută maximă, relativă și relativă maximă ale aproximantei, în acest caz.

42. Să se calculeze suma :

$$0{,}816 + 1{,}173 + 1{,}6391 + 1{,}2928 + 0{,}7710 + 1{,}9768 + 2{,}1098$$

cu două zecimale sigure și apoi să se calculeze eroarea absolută maximă știind că pentru fiecare număr eroarea absolută este indicată de ultima lui zecimală.

43. Să se calculeze suma

$$759{,}43578 + 0{,}0072376 + 1{,}0051429$$

cu o eroare absolută de $\frac{1}{10^4}$. Pentru cazul în care erorile relative ale termenilor sumei sînt $\frac{1}{10^5}$, $\frac{1}{10^6}$ și respectiv $\frac{1}{10^7}$, să se calculeze eroarea relativă maximă a sumei.

44. Dacă suma numerelor

$$926{,}4598 + 1{,}5737983 + 0{,}027936$$

are eroarea absolută maximă $\frac{2}{10^3}$, cît pot fi erorile absolute ale numerelor ? Dar dacă eroarea relativă a sumei este $\frac{1}{10^4}$, care pot fi erorile relative ale termenilor ?

45. Să se calculeze

$$\frac{3}{4} + \frac{4}{5} + \frac{5}{6} + \frac{6}{7} + \frac{7}{8},$$

cu o eroare de $\frac{1}{10^3}$.

46. Să se calculeze cu exactitate și cu aproximație diferența

$$1{,}4823 - 0{,}265.$$

Care sînt erorile care se produc ?

47. Să se calculeze cu trei cifre sigure produsul

$$85{,}279432 \cdot 7{,}36972 \cdot 0{,}2459704.$$

48. Un mobil se deplasează cu viteza de 34,15 km/h timp de 2,45 ore. Să se determine eroarea relativă și numărul de cifre sigure ale aproximantei care dă spațiul parcurs de mobil.

49. Într-o mișcare circulară pe un cerc cu raza π un mobil se deplasează cu o viteză constantă $v = 2{,}71828$ în timpul t dat de rădăcina pozitivă a ecuației $x^2 + 4x - 1 = 0$. Să se calculeze eroarea relativă și numărul de cifre sigure ale aproximantei care dă spațiul parcurs de mobil cînd raza cercului, viteza mobilului și timpul sînt date cu trei cifre sigure.

50. Să se calculeze erorile absolute și relative ale numărului π cînd se iau următoarele aproximante

$$3\ ;\ 3{,}14\ ;\ 3 + \frac{10}{7}\ ;\ \frac{355}{113}.$$

51. Să se calculeze volumul sferei

$$V = \frac{4}{3}\pi R^3$$

știind că $R = 0{,}5 + \alpha$ cu $|\alpha| \leqslant \frac{1}{10^3}$.

52. Să se calculeze cu aproximație de $\frac{1}{10^4}$ numărul

$$\frac{1}{1\cdot 3} + \frac{1}{1\cdot 3\cdot 5} + \frac{1}{1\cdot 3\cdot 5\cdot 7} + \frac{1}{1\cdot 3\cdot 5\cdot 7\cdot 9}.$$

53. Numerele 863,4562 și 0,0725 sînt date cu o eroare absolută maximă de $\frac{1}{10^4}$. Care este eroarea absolută maximă a diferențelor lor? Dar eroarea relativă maximă?

54. Eroarea absolută maximă a diferenței,

$$53\,978{,}47978 - 3{,}27544,$$

este de $\frac{1}{10^5}$. Care pot fi erorile absolute maxime ale termenilor?

55. Să se stabilească eroarea relativă maximă a produsului $123{,}4792 \cdot 53{,}797 \cdot 5\,729{,}37$ știind că factorii au respectiv erorile relative indicate de ultima lor zecimală.

56. Cîte cifre sigure are produsul:

$$0{,}75397 \cdot 1{,}64285 \cdot 26{,}37492,$$

știind că fiecare dintre factori are toate cifrele sigure?

57. Să se calculeze cu două cifre sigure produsul:

$$0{,}75059732 \cdot 0{,}003579472 \cdot 0{,}400297594.$$

58. Să se calculeze cu primele două zecimale sigure produsul:

$$32{,}125472 \cdot 16{,}255463,$$

știind că unul din factori are primele trei zecimale sigure.

59. Să se stabilească eroarea relativă maximă a cîtului,

$$15{,}47983 : 3{,}12142,$$

știind că erorile relative ale deîmpărțitului și împărțitorului sînt date de ultimele lor zecimale.

60. Să se stabilească numărul de cifre sigure ale cîtului,

$$69{,}5793 : 23{,}1821,$$

știind că deîmpărțitul și împărțitorul au fiecare ultima zecimală cifră îndoielnică.

61. Să se calculeze cîtul,

$$0{,}849732 : 0{,}4235794,$$

avînd cel puțin primele două cifre sigure.

62. Să se calculeze cîtul,

$$56{,}379 : 28{,}378,$$

cu prima zecimală sigură știind că unul din factori are cel puțin două zecimale sigure.

63. Să se calculeze:

$(24{,}5679436)^2$ astfel ca rezultatul să aibă prima zecimală sigură.

III.14. Operații logice

III.14.1. Introducere

Ați observat desigur că analiza și rezolvarea oricărei probleme se reduc, în ultimă instanță, la o înlănțuire de afirmații (propoziții sau fraze) care ne permit să tragem anumite concluzii.

De multe ori credem că am gîndit corect, că am judecat bine, dar lucrurile nu stau așa.

Astfel, plecînd de la afirmația : „oamenii înalți sînt buni sportivi" mulți se grăbesc să conchidă că „oamenii mici de statură nu sînt buni sportivi" ceea ce nu este adevărat.

Sau din propozițiile :

„Toți oamenii în vîrstă sînt înțelepți"

„Badea Ilie este înțelept"

să se deducă „Badea Ilie este în vîrstă", afirmație incorectă, deoarece cele două propoziții nu ne permit să apreciem vîrsta lui Badea Ilie.

Un alt tip de greșeală de logică frecventă este negarea incorectă a propozițiilor.

De exemplu, se obișnuiește să se nege propoziția „Ionescu este deștept" prin „Ionescu este prost", ceea ce nu este corect din punct de vedere al logicii, deoarece negația unei propoziții de tipul „A este B" este „A nu este B", și nu „A este non-B".

Și în raționamentele matematice pot apărea greșeli de logică. Astfel, este greșit ca din afirmația „dacă un triunghi are 2 unghiuri congruente atunci el este isoscel" să se deducă afirmația „dacă un triunghi are două unghiuri diferite atunci el nu este isoscel".

Se pune atunci întrebarea : cum putem avea certitudinea că am raționat corect, că am dedus concluzii adevărate și că modul în care le-am obținut din informațiile inițiale este cel corect ?

Răspunsul la această întrebare ni-l dă *logica — știința care descoperă și formulează legile gîndirii corecte.*

Logica s-a dezvoltat încă din antichitate, fiind fondată în secolul al IV-lea î.e.n. de Aristotel. Apropiată ca spirit atît de filozofie cît și de matematică, logica a beneficiat de aportul ambelor științe. Începînd din secolul al XIX-lea unii matematicieni și filozofi au încercat să studieze logica cu metode matematice. Ideea era să se realizeze și în logică acel salt calitativ reprezentat în matematică de trecerea de la rezolvarea aritmetică la rezolvarea algebrică. Pentru aceasta propozițiile au fost notate cu simboluri[1], cu ele efectuîndu-se operații respectînd anumite reguli. Ideea calculului logic, inițial formulată de Leibniz[2], a fost preluată de A. De Morgan[3] și G. Boole[4] care au pus bazele algebrei logicii.

Astfel, logica a devenit mai riguroasă și mai ușor de aplicat, nu numai în viața de toate zilele și în rezolvarea problemelor specifice diferitelor domenii științifice, dar și în dezvoltarea unor discipline cu pronunțat caracter practic, ca de pildă : programarea calculatoarelor, automatizări bazate pe circuite cu contacte și relee sau pe circuite electronice etc.

[1] Logica matematică se mai numește și logică simbolică.

[2] G. W. Leibniz (1646—1716), matematician și filozof german.

[3] Augustus De Morgan, matematician englez (1806—1871).

[4] George Boole, matematician și logician irlandez (1815—1864).

La noi în țară, o contribuție deosebită în dezvoltarea logicii matematice au avut-o oamenii de știință Gr. C. Moisil, E. Mihăilescu, O. Onicescu, A. Dumitriu, M. Tîrnoveanu, școala românească de logică fiind cunoscută în întreaga lume.

III.14.2. Calculul propozițiilor

Deoarece obiectul logicii este studiul formelor corecte de raționament, iar un raționament constă dintr-o înlănțuire de judecăți, în logică interesează modul în care se leagă între ele propozițiile și în primul rînd, cum plecînd de la unele propoziții (considerate ca „materie primă") putem construi alte propoziții. Elementele de construcție se numesc „propoziții simple" (de exemplu : „Mă pasionează calculatoarele", „Studiez informatica").

Propozițiile formate din propoziții simple, cu ajutorul unor cuvinte de legătură, se numesc propoziții compuse (de pildă: „Mă pasionează calculatoarele și studiez informatica", „Dacă mă pasionează calculatoarele, atunci studiez informatica"; cuvintele de legătură fiind „și", „Dacă... atunci..."). Cu aceleași propoziții simple se pot forma propoziții compuse diferite.

Partea logicii matematice care studiază legăturile dintre propoziții, fără a ține seama de structura internă a propozițiilor simple, se numește logica propozițiilor sau calcul propozițional.

Urmărind tratarea logicii cu mijloace matematice, vom reprezenta simbolic propozițiile prin formule care seamănă ca structură cu cele algebrice.

Vom nota propozițiile (simple sau compuse) cu litere mari ale alfabetului latin. De exemplu, notăm propozițiile simple : „Citesc o carte interesantă" cu P, „La T.V. este un film în premieră" cu Q, „Stau acasă" cu litera R, iar propoziția compusă : „Dacă citesc o carte interesantă sau la T.V. este un film în premieră, atunci stau acasă" cu A. Observăm că, de fapt, formula A este „Dacă P sau Q atunci R". Adevărul sau falsitatea unei propoziții se numește valoarea sa logică (sau valoarea de adevăr) și se notează cu $\mathcal{A}$, respectiv $\mathcal{F}$.

Întrucît aceeași propoziție simplă poate fi, după împrejurări, adevărată sau nu, în logica matematică se face abstracție de sensul său. Cu alte cuvinte, vom presupune că nu știm care este valoarea sa de adevăr ci doar că aceasta poate fi ori $\mathcal{A}$ ori $\mathcal{F}$.

La lecțiile de algebră s-au prezentat principalele operații logice, de aceea le vom considera cunoscute, ele fiind recapitulate în tabelul III.3*. Cu ajutorul acestor operații se pot construi propoziții compuse din propoziții simple.

* În tabel figurează pe lîngă cele cinci operații logice elementare și disjuncția exclusivă, datorită frecvențelor aplicații în care apare (circuite logice, programare).

Tabelul III.3

P	$\neg P$
$\mathcal{A}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{A}$

Negația unei propoziții P este adevărată dacă propoziția P este falsă și este falsă dacă propoziția P este adevărată.

P	Q	$P \wedge Q$
$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$

Conjuncția propozițiilor P și Q este adevărată doar dacă ambele propoziții P și Q sînt adevărate și este falsă cînd cel puțin una din propoziții P și Q este falsă.

P	Q	$P \vee Q$
$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$

Disjuncția propozițiilor P și Q este adevărată dacă cel puțin una din propozițiile P și Q este adevărată și este falsă numai cînd ambele propoziții sînt false.

P	Q	$P \vee Q$
$\mathcal{A}$	$\mathcal{A}$	$\mathcal{F}$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$

Disjuncția exclusivă a propozițiilor P și Q este adevărată numai dacă una din propozițiile P și Q este adevărată și este falsă cînd propozițiile au aceeași valoare logică.

P	Q	$P \Rightarrow Q$
$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{A}$

Implicația propoziției Q prin propoziția P este falsă numai atunci cînd P este adevărată și Q este falsă ; în toate celelalte cazuri ea este adevărată.

P	Q	$P \Leftrightarrow Q$
$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{A}$

Echivalența propozițiilor P și Q este adevărată atunci cînd P și Q au aceeași valoare logică și este falsă cînd P și Q au valori logice diferite.

La rîndul lor propozițiile compuse se pot lega între ele cu propoziții simple și se obțin noi propoziții.

În evaluarea unei propoziții compuse ordinea de efectuare a operațiilor este : $\neg$, $\wedge$, $\vee$, $\forall$, $\Rightarrow$, $\Leftrightarrow$. Pentru a indica altă ordine de efectuare a operațiilor se folosesc paranteze.

Astfel, formula : $P \wedge \neg P \vee Q$ este diferită de $P \wedge \neg(P \vee Q)$. Sau, de exemplu, formula :

$$P \Rightarrow Q \vee \underbrace{\neg P}_{1} \wedge R$$
$$\underbrace{\qquad\qquad}_{2}$$
$$\underbrace{\qquad\qquad\qquad}_{3}$$
$$\underbrace{\qquad\qquad\qquad\qquad}_{4}$$

este diferită de formula :

$$(P \Rightarrow Q) \vee \underbrace{\neg P}_{1} \wedge R.$$
$$\underbrace{\qquad\qquad}_{3} \quad \underbrace{\qquad}_{2}$$
$$\underbrace{\qquad\qquad\qquad\qquad}_{4}$$

Cu ajutorul tabelelor de adevăr putem determina valorile logice ale oricărei formule compuse, ținînd seama de toate combinațiile posibile de valori logice ale componentelor.

Fie formula : $A = P \wedge Q \vee P \wedge \neg Q$.

Aceasta are drept componente pe P și Q, legate prin conectorii $\wedge$, $\vee$, $\neg$. Ordinea de efectuare a operațiilor arată că întîi se face negația lui Q, apoi conjuncțiile $P \wedge Q$ și $P \wedge \neg Q$, iar la urmă disjuncția acestor două formule.

Pentru scrierea tabelului completăm întîi coloanele P și Q (corespunzătoare formulelor simple din care este compusă formula A), cu toate combinațiile posibile de valori logice. Fiecare formulă simplă avînd numai două valori logice posibile și n fiind numărul formulelor simple care intervin într-o formulă compusă, tabelul va avea 2^n linii, deoarece cu n elemente care pot lua două valori diferite, se pot forma 2^n combinații de valori distincte. Rezultă că tabelul de adevăr al formulei A va avea $2^2 = 4$ linii.

Se completează pe rînd cîte o linie. Fiecare coloană corespunde unei propoziții, obținute printr-una din cele cinci operații logice, din propoziții ale căror valori logice figurează în coloane precedente. Valoarea logică a propoziției se determină conform tabelului de adevăr al operației în cauză.

Evaluarea corectă a formulelor cu mai multe operații logice este deosebit de importantă pentru tehnica de calcul. Vom da numai cîteva exemple.

Tabelul III.4

	P	Q	$\neg Q$	$P \wedge Q$	$P \wedge \neg Q$	$P \wedge Q \vee P \wedge \neg Q$
1	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$
2	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$
3	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$
4	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$

La calculul liniei 2, de exemplu, s-a procedat astfel:

$$\underbrace{\underbrace{P \wedge Q}_{\downarrow} \vee \underbrace{P \wedge \neg Q}_{\downarrow}}_{\downarrow}$$
$$\underbrace{\underbrace{\mathcal{A} \wedge \mathcal{F}}_{\mathcal{F}} \vee \underbrace{\mathcal{A} \wedge \mathcal{A}}_{\mathcal{A}}}_{\mathcal{A}}$$

Există circuite logice, fundamentale în realizarea echipamentelor de calcul, care realizează funcția logică de negație a disjuncției (circuite „NOR") și funcția logică de negație a conjuncției (circuite „NAND"). Acestea, la rîndul lor, intră în componența multor circuite mult mai complexe, cu ajutorul cărora se implementează funcțiile echipamentului.

Sau, printre instrucțiunile mașină ale minicalculatoarelor românești se numără două instrucțiuni logice BIC și BIS care evaluează formulele logice $P = P \wedge \neg Q$, respectiv $P = P \vee Q$, unde P și Q reprezintă biții dintr-o zonă de memorie (valorii 1 îi corespunde valoarea logică $\mathcal{A}$ și valorii 0 — valoarea logică $\mathcal{F}$). De exemplu, cu BIC se pot compara două configurații binare de 16 biți (cuvinte de memorie) și poziționa pe 0 biții cu valoarea 1 din prima configurație, cărora le corespund biți 1 în cea de-a doua:

	înainte	*după* execuția instrucțiunii BIC, A, B
A:	1111111111111111	1101001100010010
B:	0010110011101101	0010110011101101

De asemenea, în descrierea rezolvării unei probleme, poate să fie necesar să se exprime condiții compuse, care de fapt reprezintă formule logice cu mai multe operații logice.

III.14.2.1. *Diagrame Euler-Venn*

În cazul formulelor cu cel mult trei propoziții simple componente este ușor să se obțină o reprezentare geometrică pentru valoarea de adevăr a formulei respective.

Să considerăm un dreptunghi, iar în interiorul lui să delimităm o anumită zonă printr-o curbă închisă. Domeniul închis de curbă va reprezenta,

Fig. III.1

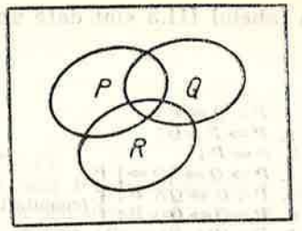


Fig. III.2

pentru o propoziție oarecare P, valoarea $\mathcal{A}$, iar zona exterioară domeniului — valoarea $\mathcal{F}$ (fig. III.1).

În cazul propozițiilor compuse, fiecare componentă va fi reprezentată printr-un domeniu. Pentru a reprezenta toate combinațiile posibile de valori logice, domeniile se aleg astfel încît să se intersecteze în toate modurile posibile (fig. III.2).

Astfel de reprezentări ale operațiilor logice se numesc diagrame Euler-Venn[1].

Pentru a descrie o anumită operație logică, figurăm propozițiile componente și haşurăm porțiunea în care propoziția compusă rezultată are valoarea $\mathcal{A}$; porțiunea nehașurată va reprezenta valoarea $\mathcal{F}$ a propoziției (conjuncția este ilustrată de figurile III.3, *a* și III.3, *b*).

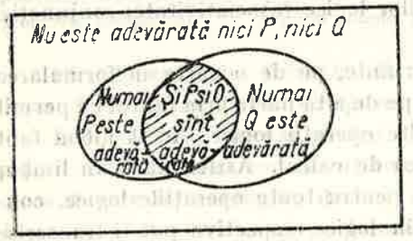


Fig. III.3, *a*

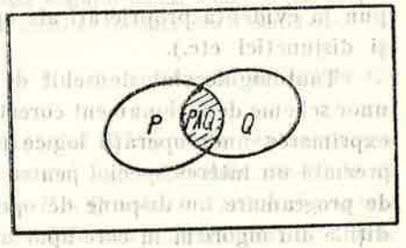


Fig. III.3, *b*

III.14.2.2. *Tautologii și contradicții*

Se știe, de la algebră, că o formulă care are valoarea logică $\mathcal{A}$, indiferent de valoarea logică a componentelor, se numește *tautologie*, sau *formulă identic adevărată*, sau încă *formulă validă*. Analog, formulele care au întotdeauna valoarea logică $\mathcal{F}$ se numesc *contradicții*, sau *formule identic false*, sau *formule nerealizabile*.

[1] După numele matematicienilor L. Euler (1707—1783) și J. Venn care au introdus astfel de reprezentări pentru mulțimi și propoziții.

În tabelul III.5 sînt date unele din cele mai utilizate tautologii.

Tabelul III.5

T. 1. $P \wedge Q \Rightarrow P$;
T. 2. $P \Rightarrow P \vee Q$;
T. 3. $P \Rightarrow P$;
T. 4. $P \Rightarrow Q \Leftrightarrow \neg Q \Rightarrow \neg P$;
T. 5. $P \wedge Q \Leftrightarrow Q \wedge P$;
T. 6. $P \vee Q \Leftrightarrow Q \vee P$; } (comutativitatea pentru $\wedge$ și $\vee$);
T. 7. $P \wedge (Q \wedge R) \Leftrightarrow (P \wedge Q) \wedge R$;
T. 8. $P \vee (Q \vee R) \Leftrightarrow (P \vee Q) \vee R$; } (asociativitatea pentru $\wedge$ și $\vee$);
T. 9. $P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$;
T.10. $P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$; } (distributivitatea $\vee$ și $\wedge$);
T.11. $P \wedge P \Leftrightarrow P$;
T.12. $P \vee P \Leftrightarrow P$; (idempotență) ;
T.13. $P \wedge (P \vee Q) \Leftrightarrow P$;
T.14. $P \vee (P \wedge Q) \Leftrightarrow P$; (absorbția) ;
T.15. $\neg \neg P \Leftrightarrow P$;
T.16. $P \vee \neg P$; (principiul terțiului exclus) ;
T.17. $\neg (P \wedge \neg P)$; (principiul necontradicției) ;
T.18. $\neg (P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$;
T.19. $\neg (P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$; } (legile lui De Morgan) ;
T.20. $\neg (P \Rightarrow Q) \Leftrightarrow P \wedge \neg Q$; T.20′. $(P \Rightarrow Q) \Leftrightarrow \neg P \vee Q$;
T.21. $(P \Rightarrow Q) \wedge (Q \Rightarrow R) \Leftrightarrow (P \Rightarrow R)$ (tranzitivitatea implicației) ;
T.22. $(P \Leftrightarrow Q) \Leftrightarrow (P \Rightarrow Q) \wedge (Q \Rightarrow P)$.

Consultînd tabelul se observă că o mare parte din tautologiile prezentate pun în evidență proprietăți ale operațiilor logice (asociativitatea conjuncției și disjuncției etc.).

Tautologiile sînt deosebit de importante, pe de o parte în formularea unor scheme de raționament corecte, iar pe de altă parte prin faptul că permit exprimarea unor operații logice prin alte operații logice. Acest ultim fapt prezintă un interes special pentru tehnica de calcul. Astfel, dacă un limbaj de programare nu dispune de operatori pentru toate operațiile logice, condițiile din algoritm în care apar operațiile logice respective pot fi transcrise în limbaj de programare folosind proprietăți de tipul celor exprimate de tautologiile T20 + T22. Astfel, de exemplu, în FORTRAN, o condiție de tipul $A \Rightarrow B$ cu A și B variabile logice, se va codifica prin .NOT.A.OR.B (.AND., .NOT., .OR. fiind operatorii FORTRAN pentru $\wedge$, $\neg$ și $\vee$), acest limbaj neavînd un operator logic corespunzător implicației. Pe de altă parte, o situație similară poate apărea și la realizarea circuitelor logice (prin combinații de circuite „NOR" și „NAND" se pot realiza circuite logice care să descrie orice formulă logică).

Observația 1. Să considerăm formula $\neg (P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$; dacă în $\neg (P \wedge Q)$ și în $\neg P \vee \neg Q$ schimbăm pe $\wedge$ cu $\vee$ și pe $\vee$ cu $\wedge$ se vede că obținem cea de-a doua lege a lui De Morgan (T.19). Acest lucru nu este întîmplător, ci exprimă o proprietate a conjuncției și disjuncției numită dualitate. Prin duala unei formule se înțelege formula obținută prin schimbarea conectorilor $\wedge$ și $\vee$ cu $\vee$, respectiv $\wedge$. Duala unei formule are același tabel de adevăr cu negația formulei în care s-a substituit P cu $\neg P$ și invers.

Astfel, fie de exemplu formula :

$$P \wedge \neg P.$$

Duala sa este : $P \vee \neg P$.

Plecînd de la formula inițială obținem

a) negația $\neg (P \wedge \neg P)$

b) substituția P cu $\neg P$ și $\neg P$ cu P ne dă $\neg (\neg P \wedge P)$,

ceea ce revine la $\neg (\neg P) \vee \neg P$, adică $P \vee \neg P$. Astfel, dacă A și B sînt formule compuse construite din formule simple sau negații ale formulelor simple, numai cu ajutorul operațiilor $\wedge$ și $\vee$ și dacă A' și B' sînt formule obținute din A și B prin schimbarea lui $\wedge$ cu $\vee$ și a lui $\vee$ cu $\wedge$, atunci :

1) dacă $\neg A$ este tautologie și A' este tautologie ;
2) dacă A este tautologie și $\neg A'$ este tautologie ;
3) dacă $A \Leftrightarrow B$ este tautologie și $A' \Leftrightarrow B'$ este tautologie ;
4) dacă $A \Rightarrow B$ este tautologie și $B' \Rightarrow A'$ este tautologie.

Observația 2. În cazul în care o tautologie se exprimă printr-o echivalență (ca de pildă T.19 : $\neg (P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$) se spune că formulele legate prin semnul „$\Leftrightarrow$" sînt logic echivalente (au aceleași valori logice). Într-o formulă compusă, o anumită componentă poate fi înlocuită printr-o formulă logic echivalentă, fără ca valoarea logică a formulei compuse să se modifice. Astfel,

$$P \wedge Q \vee \neg (\neg R \wedge \neg S) \Leftrightarrow P \wedge Q \vee R \vee S$$ în virtutea lui T.19 și T.15.

Observația 3. Pentru a constata că o formulă este identic adevărată nu este totdeauna necesar să construim tabelul de adevăr (sau diagrama Euler-Venn) pentru toate propozițiile simple componente (acest lucru devenind și foarte complicat pentru formule cu multe componente). Dacă în tabelul de adevăr al unei formule obținem numai valori $\mathcal{A}$ considerînd propozitii componente, nu neapărat simple, putem fi siguri că formula este o tautologie.

Fie, de exemplu, formula (1) $P \wedge Q \Rightarrow P \wedge Q$, care este de forma (2) $A \Rightarrow A$ cu A propoziția compusă $P \wedge Q$.

Din tabelul de adevăr pentru $A \Rightarrow A$ (tabelul III.6) unde pentru propoziția compusă A am considerat cele două valori logice posibile, se vede că $A \Rightarrow A$ este o tautologie, deci și $P \wedge Q \Rightarrow P \wedge Q$ este o tautologie (acest lucru se poate verifica și pe tabelul III.7, unde am calculat întîi valorile

Tabelul III.6

A	$A \Rightarrow A$
$\mathcal{A}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{A}$

Tabelul III.7

P	Q	$P \wedge Q$	$P \wedge Q \Rightarrow P \wedge Q$
$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{A}$

pentru $P \wedge Q$, adică A și plecînd de la acestea valorile implicației $P \wedge Q \Rightarrow P \wedge Q$).

Să observăm acum că tabelul de adevăr pentru $A \Rightarrow A$ și cel pentru $P \Rightarrow P$ (tabelul III.8) cu P propoziție simplă diferă numai prin notații. Astfel

pentru a arăta că $A \Rightarrow A$ este tautologie este suficient să arătăm că $P \Rightarrow P$ este tautologie și apoi să înlocuim pe P cu A în $P \Rightarrow P$.

Tabelul III.8

P	$P \Rightarrow P$
$\mathcal{A}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{A}$

Tabelul III.9

P	Q	$\neg Q$	$P \vee \neg Q$
$\mathcal{A}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$

Raționînd ca în exemplul precedent se poate demonstra următorul rezultat:

Teorema 1 (teorema substituției). **Fie A o propoziție compusă din propozițiile simple $P_1, P_2, \ldots, P_n$ și B o propoziție compusă obținută prin înlocuirea în A a propozițiilor simple $P_1, \ldots, P_n$ cu propozițiile compuse $Q_1, \ldots, Q_n$. Dacă A este tautologie atunci și B este tautologie.**

Această teoremă este importantă deoarece permite:

a) să afirmăm că lista de tautologii T.1—T.22 este valabilă pentru P Q și R formule oarecare;

b) să extindem lista de tautologii.

Trebuie să observăm că reciproca teoremei nu este adevărată; dacă A nu este tautologie, B poate sau nu să fie o tautologie. Astfel, dacă A este formula $P \vee Q$ (știm că disjuncția nu este o tautologie) și înlocuim pe Q cu $\neg P$ obținem $B = P \vee \neg P$ care este o tautologie (principiul terțiului exclus). Înlocuind însă pe Q cu $\neg Q$ obținem $B = P \vee \neg Q$, care nu mai este tautologie (vezi tabelul III.9).

Există și alte rezultate importante care ne permit să stabilim că anumite formule sînt tautologii.

Fie următoarea:

Teorema 2. **Dacă A este tautologie și dacă $A \Rightarrow B$ este tautologie atunci și B este tautologie.**

A demonstra această teoremă revine la a determina în condițiile date valorile logice ale lui B (tabelul III.10).

Tabelul III.10

A	B	$A \Rightarrow B$
$\mathcal{A}$	?	$\mathcal{A}$
$\mathcal{A}$	?	$\mathcal{A}$
$\mathcal{A}$	?	$\mathcal{A}$
$\mathcal{A}$	?	$\mathcal{A}$

Tabelul III.11

P	Q	$P \Rightarrow Q$
$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{A}$

Din tabelul de adevăr al implicației (tabelul III.11) se vede că $P \Rightarrow Q$ are valoarea $\mathcal{A}$ pentru $P = \mathcal{A}$ numai dacă și Q este $\mathcal{A}$. Substituind pe A lui P și pe B lui Q, ajungem la concluzia că B este tautologie.

Să utilizăm această teoremă și teorema substituției pentru a arăta că $P \vee \neg P \vee Q$ este o tautologie, fără a construi tabelul de adevăr.

Într-adevăr să înlocuim în $P \Rightarrow P \vee Q$ (T.2) pe P cu $P \vee \neg P$; obținem $P \vee \neg P \Rightarrow P \vee \neg P \vee Q$. Aceasta este o tautologie, conform teoremei substituției. Pe de altă parte, deoarece $P \vee \neg P$ este o tautologie, din teorema 2 rezultă că și $P \vee \neg P \vee Q$ este o tautologie c.c.t.d.

III.14.2.3. *Raționament*

Pentru rezolvarea unei probleme, formulăm mai multe propoziții care exprimă judecăți legate între ele.

Să urmărim următoarele propoziții:

„Dacă plouă, Andrei își ia umbrela".

„Plouă".

„Așadar, Andrei își ia umbrela".

Observăm că ultima propoziție apare ca o urmare a celorlalte două; prima propoziție ne spune ce face Andrei dacă plouă; a doua propoziție precizează: „Plouă"; acum știm ce face Andrei și anume „își ia umbrela".

O înlănțuire de judecăți (propoziții) în care plecînd numai de la anumite cunoștințe — consemnate într-un număr de propoziții numite premise — se ajunge la o cunoștință nouă — exprimată printr-o propoziție numită concluzie — se numește *raționament.*

Astfel, înlănțuirea de judecăți din exemplul precedent este un raționament, avînd drept premise: „Dacă plouă, Andrei își ia umbrela" și „Plouă", iar drept concluzie „Așadar Andrei își ia umbrela".

Un raționament este corect dacă și numai dacă concluzia derivă din premise.

Raționamentul din exemplul anterior este corect, la fel și raționamentul:

Premise: „$x + y = 10$"

„$x = 3$"

Concluzie: „$y = 7$".

Nu trebuie confundată corectitudinea unui raționament cu adevărul concluziei. Astfel, raționamentul:

Premise: „Toți copacii sînt păsări"

„Toate păsările pot înota"

Concluzie: „Toți copacii pot înota"

este corect, dar concluzia este falsă, decurgînd din premise false.

Raționamentul :

Premise : „Toate păsările cîntă" ;

„Toate rîndunelele cîntă".

Concluzie : „Toate rîndunelele sînt păsări",

este incorect, deși concluzia este adevărată. Raționamentul este incorect deoarece concluzia nu derivă din premise.

Să considerăm acum ca premise ale unui raționament propozițiile :

„Dacă măsura $\sphericalangle A$ = măsura $\sphericalangle B$, triunghiul ABC este isoscel"

„Dreptele D_1 și D_2 sînt paralele".

Din aceste două propoziții nu putem desprinde nici o concluzie, deci nu orice propoziții considerate drept premise ne conduc la o concluzie.

Așadar logica, avînd menirea să descopere și să formuleze legile gîndirii corecte, adică formele corecte de raționament, trebuie să răspundă la următoarea întrebare :

Avînd o mulțime de propoziții adevărate, considerate drept premise cum putem ajunge la concluzii și care sînt acestea?

III.14.2.4. *Tipuri de raționamente*

Pe baza proprietăților operațiilor logice se pot construi raționamente corecte. Acestea constituie „modele" de raționament valabile în orice teorie în care despre o anumită propoziție putem spune că este fie adevărată, fie falsă, dar nu ambele, adică în orice domeniu în care este valabil principiul bivalenței.

După cum s-a mai precizat, în cele ce urmează, se va vedea că diferitele tipuri de raționamente valabile se bazează pe tautologii.

III.14.2.4.1. *Raționament prin modus ponens*

Cel mai simplu tip de raționament și, în același timp, elementul pe care se fundamentează toate tipurile de raționamente este următorul : „dacă P și în același timp $P \Rightarrow Q$, atunci Q". Observăm că acest mod de gîndire ne este firesc, deoarece sîntem obișnuiți cu el atît din matematică, cît și din viața de toate zilele. De asemenea, să ne amintim că pentru a demonstra teorema 2 am arătat că dacă P este adevărată și $P \Rightarrow Q$ este adevărată, atunci Q este adevărată.

Un astfel de raționament se numește modus ponens, deci, dacă P și $P \Rightarrow Q$ sînt propoziții adevărate atunci și Q este adevărată.

P și $P \Rightarrow Q$ sînt premisele raționamentului, iar Q concluzia. Acest lucru se scrie :

$$\frac{\begin{array}{l} P \\ P \Rightarrow Q \end{array}}{Q.}$$

Astfel, de exemplu, considerînd adevărate propozițiile :

„Dacă mă scol devreme, nu întîrzii la școală"

și

„Mă scol devreme"

prin modus ponens rezultă că este adevărată și propoziția :

„Nu întîrzii la școală".

Sau, considerînd adevărate propozițiile :

„$x = 2y$" și „$x = 2y \Rightarrow x$ este divizibil cu 2", atunci este adevărată și propoziția „x este divizibil cu 2".

Vom da în continuare exemple și de alte tipuri de raționamente :

III.14.2.4.2. *Raționament prin contrapoziție (modus tollens)*

Dacă propozițiile $P \Rightarrow Q$ și $\neg Q$ sînt adevărate, atunci este adevărată și propoziția $\neg P$; adică :

$$\frac{\begin{array}{l} P \Rightarrow Q \\ \neg Q \end{array}}{\neg P.}$$

Într-adevăr, conform T.20', $P \Rightarrow Q$ este logic echivalentă cu $\neg P \vee Q$, deci rezultă că dacă $P \Rightarrow Q$ este adevărată atunci și $\neg P \vee Q$ este adevărată ; $\neg Q$ fiind adevărată, Q este falsă. Deci pentru ca disjuncția $\neg P \vee Q$ să fie adevărată trebuie ca $\neg P$ să fie adevărată (rezultă din tabelul de adevăr al disjuncției).

Fie, de exemplu, $P \Rightarrow Q$ teorema de geometrie :

„Dacă un triunghi ABC este isoscel, atunci triunghiul ABC are două unghiuri congruente". Atunci $\neg Q$ este propoziția : „Triunghiul ABC nu are două unghiuri congruente" pe care o considerăm adevărată. Putem afirma imediat că $\neg P$ este adevărată, adică : „triunghiul ABC nu este isoscel".

III.14.2.4.3. *Raționament prin reducere la absurd*

Fie A o formulă despre care vrem să arătăm că este adevărată. Pentru aceasta vom presupune că A este falsă, deci că de fapt este adevărată formula $\neg A$.

Pornind de la această presupunere deducem o anumită propoziție $\neg B$, care este în contradicție cu un adevăr B cunoscut (B este întotdeauna adevărată în teoria considerată), ceea ce este absurd. Rezultă că $\neg A$ este falsă, deci A este adevărată.

Formalizat, acest tip de raționament se scrie :

$$\frac{\begin{array}{l} \neg A \Rightarrow B \\ \neg A \Rightarrow \neg B \end{array}}{A}$$

și poartă numele de raționament prin reducere la absurd.

Acesta este un raţionament corect. Într-adevăr, B fiind adevărată şi $\neg A \Rightarrow B$ este adevărată. Dar şi $\neg A \Rightarrow \neg B$ este adevărată. Aceste două implicaţii sînt însă simultan adevărate numai atunci cînd $\neg A$ este falsă, după cum rezultă din tabelele de adevăr corespunzătoare (tabelele III.12 şi III.13), adică numai atunci cînd A este adevărată.

Tabelul III.12

A	$\neg A$	B	$\neg A \Rightarrow B$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$

Tabelul III.13

A	$\neg A$	B	$\neg B$	$\neg A \Rightarrow \neg B$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$
$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{F}$
$\mathcal{F}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{A}$

Observăm că la acelaşi rezultat se putea ajunge folosind raţionamentul prin modus tollens.

Într-adevăr, $\neg A \Rightarrow \neg B$ este adevărată şi B este adevărată. Dar $B \Leftrightarrow \neg\neg B$ (conform T.15). Luînd drept P pe $\neg A$ şi drept Q pe $\neg B$, rezultă imediat prin modus tollens $\neg\neg A$ adevărată, deci A adevărată.

Raţionamentul prin reducere la absurd este foarte des întîlnit în geometrie.

Exemplu:

Să demonstrăm prin reducere la absurd următoarea teoremă din geometria euclidiană plană.

A: „Două drepte distincte D_1 şi D_2, paralele cu a treia D_3, sînt paralele între ele" sau

$$A: \text{„}D_1 \parallel D_3 \wedge D_2 \parallel D_3 \Rightarrow D_1 \parallel D_2\text{"}.$$

Deci A este de forma $P \wedge Q \Rightarrow R$.

Dar $P \wedge Q \Rightarrow R \Leftrightarrow \neg(P \wedge Q) \vee R$ şi deci

$\neg(P \wedge Q \Rightarrow R) \Leftrightarrow \neg(\neg(P \wedge Q) \vee R)$ şi aplicînd una din legile lui De Morgan obţinem: $\neg A \Leftrightarrow \neg\neg(P \wedge Q) \wedge \neg R$ adică

$P \wedge Q \wedge \neg R$ sau $D_1 \parallel D_3 \wedge D_2 \parallel D_3 \wedge D_1 \nparallel D_2$.

Rezultă că dacă $\neg A$ este adevărată, atunci şi $D_1 \nparallel D_2$ este adevărată. Dar dacă $D_1 \nparallel D_2$, atunci ele se intersectează într-un punct O. Dar $D_1 \parallel D_3$ şi $D_2 \parallel D_3$ sînt adevărate, deci punctul O nu aparţine lui D_3 şi deci printr-un punct exterior unei drepte D_3 trec două drepte paralele cu aceasta.

Dar axioma paralelelor afirmă B: „Printr-un punct exterior unei drepte se poate duce o singură paralelă la dreapta dată". Am ajuns astfel la o contradicţie: din $\neg A$ am dedus $\neg B$, iar B este adevărată. Aşadar, presupunerea $\neg A$ este falsă: prin urmare A este demonstrată.

O altă schemă posibilă pentru raţionamentul prin reducere la absurd este următoarea:

$$\frac{\begin{array}{c} B \\ \neg A \Rightarrow \neg B \end{array}}{A.}$$

Dacă B este adevărată şi $\neg A \Rightarrow \neg B$ este adevărată, din tabelul de adevăr al implicaţiei rezultă că $\neg A$ este falsă (deoarece $\neg B$ este falsă), deci A este adevărată.

III.14.2.4.4. *Raţionament prin adjuncţiune*

Dacă propoziţiile P şi Q sînt adevărate, atunci este adevărată şi conjuncţia lor $P \wedge Q$:

$$\frac{\begin{array}{c} P \\ Q \end{array}}{P \wedge Q.}$$

Faptul că acest raţionament este corect, rezultă imediat din definiţia conjuncţiei.

Fie, de exemplu, P: „Triunghiul ABC este dreptunghic" şi Q: „Triunghiul ABC este isoscel", ambele adevărate. Rezultă $P \wedge Q$ adevărată: „Triunghiul ABC este dreptunghic şi isoscel".

III.14.2.4.5. *Raţionament folosind tranzitivitatea implicaţiei*

Dacă propoziţiile $P \Rightarrow Q$ şi $Q \Rightarrow R$ sînt adevărate atunci şi propoziţia $P \Rightarrow R$ este adevărată, adică:

$$\frac{\begin{array}{c} P \Rightarrow Q \\ Q \Rightarrow R \end{array}}{P \Rightarrow R.}$$

Acest tip de raţionament este corect, conform T.21 ($(P \Rightarrow Q) \wedge (Q \Rightarrow R)$ este adevărată conform raţionamentului prin adjuncţiune).

Fie $P \Rightarrow Q$ propoziţia: „Dacă $\alpha \subset \beta$ atunci $\alpha \cap \beta = \alpha$" şi $Q \Rightarrow R$: „Dacă $\alpha \cap \beta = \alpha$, atunci $C\beta \subset C\alpha$", ambele adevărate în teoria mulţimilor*.

Rezultă că este adevărată propoziţia $P \Rightarrow R$, adică: „Dacă $\alpha \subset \beta$, atunci $C\beta \subset C\alpha$".

III.14.2.4.6. *Trecerea de la echivalenţă la implicaţie*

Dacă propoziţia $P \Leftrightarrow Q$ este adevărată, atunci sînt adevărate şi propoziţiile $P \Rightarrow Q$ şi $Q \Rightarrow P$:

$$\frac{P \Leftrightarrow Q}{P \Rightarrow Q} \text{ şi } \frac{P \Leftrightarrow Q}{Q \Rightarrow P.}$$

Se ştie că $P \Leftrightarrow Q$ este adevărată numai dacă P şi Q au aceleaşi valori logice; dar dacă P şi Q au aceleaşi valori logice, din definiţia implicaţiei rezultă că şi $P \Rightarrow Q$ şi $Q \Rightarrow P$ sînt adevărate.

Fie $P \Leftrightarrow Q$ propoziţia „$\alpha \subset \beta$ dacă şi numai dacă orice element din α este şi element al lui β". Atunci sînt adevărate şi afirmaţiile „dacă $\alpha \subset \beta$ atunci orice element din α este şi element al lui β" şi „dacă orice element din α este şi element al lui β atunci $\alpha \subset \beta$".

* Notăm mulţimile cu litere mici ale alfabetului grec (α, β, ..., ω, ...).

III.14.2.4.7. *Trecerea de la două implicații reciproce la echivalență*

Dacă propozițiile $P \Rightarrow Q$ și $Q \Rightarrow P$ sînt adevărate atunci și $P \Leftrightarrow Q$ este adevărată :

$$\frac{\begin{array}{l} P \Rightarrow Q \\ Q \Rightarrow P \end{array}}{P \Leftrightarrow Q.}$$

Raționamentul este corect conform T.22 ($(P \Rightarrow Q) \wedge (Q \Rightarrow P)$ este adevărată conform raționamentului prin adjuncțiune).

De exemplu, fie propozițiile adevărate din teoria mulțimilor :

$P \Rightarrow Q$: „$\alpha \cup \beta = \alpha \Rightarrow \beta \subset \alpha$“ și $Q \Rightarrow P$: „$\beta \subset \alpha \Rightarrow \alpha \cup \beta = \alpha$“, rezultă adevărată propoziția $P \Leftrightarrow Q$: „$\alpha \cup \beta = \alpha \Leftrightarrow \beta \subset \alpha$“.

În exemplele anterioare am utilizat diverse tipuri de raționamente în efectuarea unor demonstrații simple. Pentru demonstrarea unor teoreme mai dificile se recurge la înlănțuirea mai multor raționamente de același tip sau diferite. Avantajul utilizării diferitelor scheme de raționament constă în scurtarea demonstrației, înlocuind mai multe afirmații printr-o concluzie îndreptățită de raționamentul considerat.

ÎNTREBĂRI ȘI EXERCIȚII

1. Ce este logica ?

2. Care din următoarele formulări sînt propoziții în sensul celor exprimate la pagina 75 ?

a) Toți oamenii sînt muncitori.

b) Ionescu este elev în clasa a X-a.

c) Cînd pisica nu-i acasă, joacă șoarecii pe masă.

d) Această navă este rapidă.

e) Ce vînt te aduce ?

f) Pătratul este romb.

g) Triunghiul are patru laturi ?

h) $5 > 3$.

i) Dacă nu vin, înseamnă că nu întîrziați.

3. Care din următoarele propoziții sînt simple și care sînt compuse ; indicați propozițiile componente.

a) Astăzi nu merg la teatru.

b) Dacă ABC este un triunghi dreptunghic, atunci pătratul ipotenuzei este egal cu suma pătratelor catetelor.

c) Ninge.

d) Nu știu să înot, deci mă pot îneca.

e) Știu logică, dar nu pot să demonstrez această teoremă.

4. Să se scrie simbolic propozițiile compuse de la exercițiul 3. Să se construiască tabelul de adevăr pentru propoziția e.

5. Să se construiască tabelele de adevăr și diagramele Euler-Venn ale următoarelor formule :

a) $\neg (P \wedge Q) \Rightarrow (P \Rightarrow Q)$.

b) $\neg (P \Leftrightarrow Q) \wedge P$.

c) $(P \Rightarrow \neg Q) \wedge P \Rightarrow Q$.

6. Să se verifice că următoarele formule (pagina 80) sînt identic adevărate :

a) T.4, T.11, T.12 — cu ajutorul tabelelor de adevăr ;

b) T.13, T.14 — cu ajutorul diagramelor Euler-Venn.

7. Să se arate că relația „este logic echivalent“ este o relație de echivalență în mulțimea propozițiilor.

8. Să se arate că dacă $P \Leftrightarrow Q$ atunci și $\neg P \Leftrightarrow \neg Q$ și $P \wedge R \Leftrightarrow Q \wedge R$ și $P \vee R \Leftrightarrow Q \vee R$.

9. Să se exprime disjuncția prin negație și conjuncție.

10. Să se exprime conjuncția prin negație și disjuncție.

11. Să se exprime implicația prin negație și conjuncție.

12. Să se exprime conjuncția prin negație și implicație.

13. Să se exprime disjuncția prin negație și implicație.

14. Să se arate, fără a folosi tabele de adevăr, că următoarele formule sînt tautologii :

a) $(A \Leftrightarrow B) \Leftrightarrow (A \wedge B) \vee (\neg A \Leftrightarrow \neg B)$;

b) $\neg (A \Leftrightarrow B) \Leftrightarrow (A \wedge \neg B) \vee (\neg A \wedge B)$;

c) $\neg\neg(\neg A \vee \neg B) \Leftrightarrow \neg(\neg\neg A \wedge \neg\neg B)$.

15. Să se arate că oricare ar fi formula A, ca este echivalentă logic cu o formulă R în care negația $\neg$ nu se aplică decît propozițiilor simple.

16. Să se scrie simbolic următorul raționament :

Dacă m și n sînt numere pare, atunci $m^2 + n^2$ este par.

Dar $2a$ și $2b$ sînt pare.

Deci $4a^2 + 4b^2$ este par.

17. Să se demonstreze prin reducere la absurd teorema „Dacă $a \cdot b = 0$ atunci $a = 0$ sau $b = 0$“.

18. Să se scrie simbolic următorul enunț :

„Se știe că dacă una din cele trei persoane care discută spune ceva adevărat, atunci toate celelalte afirmații ale sale vor fi adevărate, iar dacă spune ceva fals, tot ce va spune este fals. Prima persoană face o afirmație, pe care nu o știm. Apoi cea de a doua declară : „A spus că a spus adevărul“ și „De fapt, a spus adevărul“. Cea de a treia persoană spune : „Nu, cel care a vorbit primul a mințit“.

Se poate deduce care persoane au spus adevărul ?

19. Dați exemple de raționamente prin trecere de la echivalență la implicație, folosite în rezolvarea problemelor de geometrie.

20. Să se arate că următoarele formule sînt tautologii :

a) $(P \vee Q) \Leftrightarrow ((P \wedge Q) \vee (P \wedge \neg Q) \vee (\neg P \wedge Q))$.

b) $P \Leftrightarrow (P \vee (Q \wedge \neg Q))$.

c) $P \Leftrightarrow (P \vee (P \wedge Q))$.

d) $((P \wedge Q) \Leftrightarrow (P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee Q))$.

Să se scrie dualele formulelor b) și c).

III.14.3. Calculul predicatelor

III.14.3.1. *Calculul predicatelor ca dezvoltare a calculului propozițiilor*

Pentru început, să ne amintim cîteva elemente de teoria mulțimilor. Se știe că putem reprezenta o mulțime în două feluri. Un prim procedeu constă în enumerarea elementelor din care este alcătuită mulțimea : se poate vorbi de „mulțimea elevilor : Ionescu Ion, Georgescu Dan, Popescu Dinu", sau de „mulțimea numerelor : -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5". Acest lucru se scrie de obicei punînd elementele între acolade :

$A = \{$Ionescu Ion, Georgescu Dan, Popescu Dinu$\}$;

$B = \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$.

Procedeul devine incomod dacă numărul elementelor este foarte mare și este de neutilizat, în cazul mulțimilor infinite.

Există și un alt mod de a defini o mulțime și anume prin specificarea unei proprietăți pe care o au toate elementele din mulțimea considerată ; astfel se poate vorbi de „mulțimea tuturor premianților din clasa a IX-a" sau de „mulțimea tuturor numerelor întregi x care verifică inegalitățile $-5 \leqslant x \leqslant 5$" (este chiar mulțimea B precizată anterior) ; sau de „mulțimea cărților dintr-o bibliotecă". În acest mod, și numai așa, se pot defini și mulțimi cu un număr infinit de elemente, de pildă : „mulțimea numerelor întregi", sau „mulțimea numerelor întregi pare", sau „mulțimea cercurilor cu aria 4".

Observăm că acest procedeu stabilește o legătură între mulțimi și proprietăți din logica matematică : se precizează o anumită mulțime de obiecte (mulțimea elevilor dintr-o anumită clasă sau mulțimea numerelor întregi, de exemplu) și apoi se enunță o propoziție care este adevărată pentru toate elementele mulțimii considerate și numai pentru acestea de exemplu, „este un elev foarte bun" sau „are aptitudini sportive" etc. Mulțimea α a tuturor elementelor din mulțimea totală τ (τ poate fi, de exemplu, mulțimea elevilor sau mulțimea numerelor) care au proprietatea enunțată printr-o propoziție A se numește mulțimea de adevăr a propoziției A.

Fie de exemplu τ mulțimea numerelor întregi, iar α mulțimea numerelor întregi pare. Acest lucru se poate scrie :

$$\alpha = \{x \mid x \text{ este număr par}\}.$$

Sau dacă β este mulțimea numerelor întregi cuprinse în intervalul închis $[-5, 5]$:

$$\beta = \{x \mid -5 \leqslant x \leqslant 5\}.$$

Sau dacă γ este mulțimea perechilor ordonate de numere întregi nenegative a căror sumă este 25, deducem :

$$\gamma = \{(x, y) \mid x + y = 25,\ x \geqslant 0,\ y \geqslant 0,\ x, y \text{ întregi}\}.$$

Referitor la propozițiile „x este număr par", „$-5 \leqslant x \leqslant 5$", „$x + y = 25$", se impun două observații :

a) subiectul propoziției (despre cine se afirmă că are o anumită proprietate) nu este specificat (x sau x și y) ; se precizează numai proprietatea. Proprietatea nu se modifică, indiferent de elementul cu care se înlocuiește x sau y ;

b) subiectul nefiind precizat, nu putem atribui valoarea de adevăr sau fals unei astfel de propoziții ; propoziția devine însă adevărată sau falsă în momentul precizării subiectului.

De pildă, pentru $x = 2$, propoziția „x este număr par" este adevărată, dar este falsă pentru $x = 3$; „$x + y = 25$" este adevărată pentru $x = 7$ și $y = 18$ și este falsă pentru $x = 1$ și $y = 2$.

Se știe, de la algebră, că astfel de propoziții se numesc *predicate**.

Un predicat este o afirmație relativă la una sau mai multe variabile care are proprietatea că pentru valori specificate ale variabilelor este fie adevărată, fie falsă.

Astfel, predicatele pot fi unare, binare, ternare etc., după cum depind de una, două sau mai multe variabile.

Vom nota un predicat cu $P(x)$, $Q(x)$, $R(x)$ etc. sau $P(x, y, \ldots)$ iar prin $P(a)$, $P(3)$, $P(1, 2, \ldots)$ propoziția obținută prin specificarea valorilor variabilelor.

Un predicat este definit în momentul în care se precizează mulțimea elementelor în care variabilele iau valori (mulțimea totală τ, sau mulțimea de referință).

Mulțimea de adevăr a unui predicat este mulțimea elementelor care au proprietatea P.

---

* Spre deosebire de sensul atribuit predicatului în gramatică, în logică, el exprimînd o anumită proprietate a subiectului, va cuprinde pe lîngă predicatul gramatical și complemente, atribute etc.

Altfel spus, oricărei mulțimi de elemente τ și oricărui predicat $P(x)$ le corespunde o mulțime $\mathcal{D}$ ale cărei elemente sînt exact elementele din τ pentru care $P(x)$ este adevărat. $\mathcal{D}$ este mulțimea de adevăr a predicatului.

Afirmația anterioară constituie o axiomă din teoria mulțimilor și anume axioma specificației.

Axioma specificației determină pe $\mathcal{D}$ în mod unic. Se obișnuiește să se scrie :

$$\mathcal{D} = \{x \mid x \in \tau \wedge P(x)\}$$

sau cînd mulțimea de referință τ este evidentă :

$$\mathcal{D} = \{x \mid P(x)\}.$$

Exemple :

Fie τ mulțimea numerelor întregi :

1) Predicatul „$-5 \leqslant x \leqslant 5$" este unar și are mulțimea de adevăr $\{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$.

2) Predicatul „$x^2 - 4 = 0$", este unar : domeniul este $\{-2, 2\}$.

3) Predicatul „$4x^2 + 5x + 1 = 0$" este unar, mulțimea de adevăr fiind $\{-1\}$ (a doua rădăcină $-1/4$ nu este număr întreg, deci nu aparține mulțimii de referință).

4) Predicatul „$x^2 + 1 = 0$" este predicat unar, mulțimea sa de adevăr fiind mulțimea vidă $\emptyset$.

5) Fie acum τ mulțimea perechilor ordonate de numere întregi nenegative : predicatul „$x + y = 4$" este binar : mulțimea de adevăr a predicatului este $\{(1, 3), (2, 2), (3, 1), (4, 0), (0, 4)\}$.

6) Predicatul „$x = y + z$" este ternar : mulțimea de adevăr nu se poate preciza prin enumerare, fiind infinită (de exemplu, elemente din domeniu sînt : (1, 0, 1), (1, 1, 0), (2, 0, 2), (2, 2, 0), (2, 1, 1) etc.), dacă considerăm $\tau = Z \times Z \times Z$.

7) Fie τ mulțimea totală din teoria mulțimilor. Predicatul „$\alpha \subset \beta$" reprezintă relația de incluziune.

8) Fie τ mulțimea membrilor unei familii (fig. III.4). Predicatul „x este frate cu y" este binar și are ca mulțime de adevăr, {(Ion, Petre), (Petre, Ion), (Vlad, Dinu), (Dinu, Vlad)}.

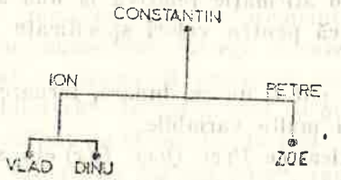


Fig. III.4

9) Fie $\tau = \tau_1 \times \tau_2$, unde τ_1 este mulțimea elevilor din clasa a IX-a și τ_2 este mulțimea elevilor din clasa a X-a de la o anumită școală. Putem considera predicatul „x are media mai mare la matematică decît y", unde $x \in \tau_1$ și $y \in \tau_2$.

Predicatele multiple (binare, ternare etc.) se numesc și relații (binare, ternare etc.).

Astfel, de exemplu, în loc să spunem că x și y au proprietatea „$x \leqslant y$", spunem „x este în relația $\leqslant$ cu y". Sau în loc de x, y are proprietatea „x este frate cu y", spunem „x este în relația, este frate cu y".

Atît predicatele unare cît și relațiile au o deosebită importanță în matematică și își găsesc multiple aplicații în cele mai diverse domenii (informatică, lingvistică, biologie etc.).

Partea logicii matematice care studiază proprietățile predicatelor precum și operațiile și raționamentele care se pot face cu ele se numește logica predicatelor sau calculul predicatelor. Logica predicatelor apare ca o extensie a logicii propozițiilor, pătrunzînd cu analiza în structura internă a propozițiilor simple, adică ținînd seama de alcătuirea lor din subiect și predicat.

III.14.3.2. *Particularizarea unui predicat. Cuantificări*

Din exemplele date în paragraful anterior se vede că un predicat este o funcție definită pe mulțimea de referință τ, al cărui codomeniu îl reprezintă valorile de adevăr ale propozițiilor rezultate prin specificarea valorilor variabilelor (fig. III.5).

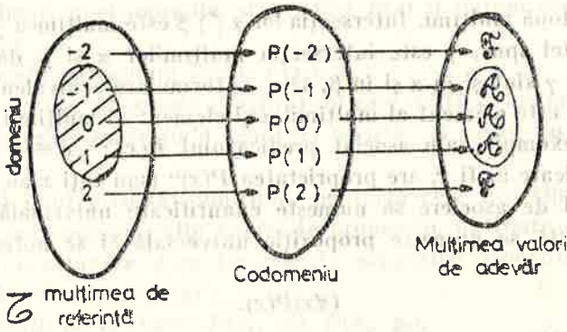


Fig. III.5

De exemplu, dacă mulțimea de referință este $\{-2, -1, 0, 1, 2\}$ și $P(x)$ este „$-1 \leqslant x < 2$", codomeniul este format din propozițiile $P(-2)$: „$-1 \leqslant -2 < 2$", $P(-1)$: „$-1 \leqslant -1 < 2$", $P(0)$: „$-1 \leqslant 0 < 2$", $P(1)$: „$-1 \leqslant 1 < 2$", $P(2)$: „$-1 \leqslant 2 < 2$". Dintre acestea $P(-1)$, $P(0)$ și $P(1)$ sînt adevărate, iar $P(2)$ și $P(-2)$ sînt false.

Deci unui predicat i se pot asocia mai multe propoziții, fiecare avînd o anumită valoare de adevăr.

În cazul exemplului precedent acest lucru se poate urmări și pe tabelul de adevăr III.14.

Tabelul III.14

x	$P(x)$
-2	$\mathcal{F}$
-1	$\mathcal{A}$
0	$\mathcal{A}$
1	$\mathcal{A}$
2	$\mathcal{F}$

Construirea tabelelor de adevăr este însă dificil de realizat pentru mulțimi de referință cu multe elemente și imposibilă pentru mulțimi infinite.

Există două metode de construire a propozițiilor asociate unui predicat $P(x)$.

Prima metodă, care de altfel a și fost utilizată, se numește particularizarea predicatului și constă în a atribui variabilelor predicatului valori efective din mulțimea de referință. Astfel unui predicat $P(x)$ îi asociem propozițiile $P(x_1)$, $P(x_2)$, ... care înseamnă „x_1 are proprietatea P“, „x_2 are proprietatea P“, iar unui predicat multiplu, de exemplu, $P(x, y)$ îi asociem propozițiile $P(x_1, y_1)$ („x_1 este în relația P cu y_1“), $P(x_2, y_2)$ („x_2 este în relația P cu y_2“) etc. Semnificația propozițiilor particulare astfel obținute depinde de elementele reprezentate de x_1, x_2, y_1, y_2. Aceste elemente se numesc variabile libere.

O altă metodă asociază predicatelor un tip special de propoziții, numite cuantificări ale predicatului, care spre deosebire de propozițiile particulare, presupun participarea parțială sau totală a elementelor din mulțimea de referință.

Există două tipuri de cuantificări:

1) *Cuantificarea universală*

Fie α și β două mulțimi. Intersecția lor $\alpha \cap \beta$ este mulțimea $\gamma = \{x | x \in \alpha$ și $x \in \beta\}$. Altfel spus, γ este intersecția mulțimilor α și β dacă și numai dacă toți x din γ sînt și în α și în β, sau, — formulare echivalentă — oricare ar fi x din γ, x este element al mulțimii α și element al mulțimii β.

În acest exemplu, am asociat predicatului $P(x)$: „$x \in \alpha$ și $x \in \beta$“ o propoziție, „oricare ar fi x, are proprietatea $P(x)$“ (sau toți x au proprietatea $P(x)$). O astfel de asociere se numește cuantificare universală, iar propoziția care rezultă se numește propoziție universală și se notează simbolic prin:

$$(\forall x)P(x).$$

Simbolul „$\forall$“ se numește *cuantor universal* sau *cuantificator universal*.

Deoarece semnificația propoziției universale nu depinde explicit de x, variabila x se numește, în acest caz, variabilă legată. Propoziția $(\forall x)P(x)$ este adevărată dacă și numai dacă toate elementele x din mulțimea de referință τ au proprietatea P și este falsă dacă pentru cel puțin un element x din τ proprietatea nu este adevărată.

În cazul unui predicat multiplu, de exemplu $P(x, y, \ldots)$ propoziția universală asociată este:

$$(\forall x)(\forall y)\ldots P(x, y, \ldots)$$

x, y, ... fiind variabilele legate.

2) *Cuantificare existențială*

Să considerăm din nou două mulțimi α și β și să presupunem că $\alpha \not\subset \beta$. Ce înseamnă acest lucru? Înseamnă că există cel puțin un element x astfel încît dacă $x \in \alpha$ atunci $x \notin \beta$.

Predicatului „dacă $x \in \alpha$ atunci $x \notin \beta$“ îi putem asocia propoziția „există cel puțin un x cu proprietatea dacă $x \in \alpha$ atunci $x \notin \beta$“. O asemenea asociere se numește cuantificare existențială, iar propoziția rezultată se numește propoziție existențială și se notează:

$$(\exists x)P(x)$$

unde „$\exists$“ reprezintă *cuantificatorul existențial*.

Și în acest caz x este variabilă legată.

Propoziția existențială este adevărată dacă și numai dacă există cel puțin un element din mulțimea de referință a predicatului pentru care proprietatea este adevărată și este falsă dacă nici un element din mulțimea de referință nu are proprietatea P.

Pentru predicate multiple $P(x, y, \ldots)$ existențiala asociată are forma:

$$(\exists x)(\exists y)\ldots P(x, y, \ldots)$$

În concluzie, unei anumite proprietăți $P(x)$ îi putem asocia trei tipuri de propoziții:

1. *Propoziții particulare* $P(x_1)$, $P(x_2)$, ..
2. *O propoziție universală* $(\forall x)P(x)$ (oricare ar fi x, avem $P(x)$).
3. *O propoziție existențială* $(\exists x)P(x)$ (există cel puțin un x pentru care avem $P(x)$).

În cazul unui predicat multiplu, pe lîngă cele trei tipuri menționate anterior se pot asocia și alte tipuri de propoziții, obținute din combinarea acestora. În continuare dăm tipurile de propoziții asociate unui predicat binar (relație binară) $P(x, y)$:

1. *Propoziții particulare:* $P(x_1, y_1)$, $P(x_2, y_2)$, ..
2. *Propoziții particulare universalizate*, obținute prin aplicarea cuantorului universal uneia dintre variabile și prin particularizarea celeilalte variabile: $(\forall x)P(x, y_1)$, $(\forall x)P(x, y_2)$, ..., care se citesc: „oricare ar fi x, avem $P(x, y_1)$“, „oricare ar fi x, avem $P(x, y_2)$“. x este variabilă legată, iar y_1, y_2, ... sînt variabile libere. Sau, $(\forall y)P(x_1, y)$, $(\forall y)P(x_2, y)$, ... cînd y este variabilă legată și x_1, x_2, ... sînt variabile libere.
3. *Propoziții particulare existențiale*, obținute analog, prin utilizarea cuantorului existențial: $(\exists x)P(x, y_1)$, $(\exists x)P(x, y_2)$, ... cu x variabilă legată și y_1, y_2, ... variabile libere și $(\exists y)P(x_1, y)$, $(\exists y)P(x_2, y)$, ... cu y variabilă legată și x_1, x_2, ..., variabile libere.
4. *O propoziție universală* $(\forall x)$ $(\forall y)P(x, y)$: „oricare ar fi x și oricare ar fi y, avem $P(x, y)$“. Observăm că semnificația propoziției nu se schimbă, dacă schimbăm ordinea în care apar variabilele cu cei doi cuantori: $(\forall y)$ $(\forall x)P(x, y)$ este propoziția „oricare ar fi y și oricare ar fi x, avem $P(x, y)$“.
5. *O propoziție existențială*: $(\exists x)$ $(\exists y)P(x, y)$: „există cel puțin un x și există cel puțin un y pentru care avem $P(x, y)$“. Semnificația acestei propoziții este aceeași cu a propoziției $(\exists y)$ $(\exists x)P(x, y)$.

6. *Două propoziții existențiale universalizate*: $(\forall x)\ (\exists y)P(x, y)$ care se citește „oricare ar fi x, există cel puțin un y, astfel încît să avem $P(x, y)$“ și $(\forall y)\ (\exists x)P(x, y)$, care se citește „oricare ar fi y, există cel puțin un x, astfel încît să avem $P(x, y)$“.

7. *Două propoziții universale existențializate*: $(\exists x)\ (\forall y)P(x, y)$, adică „există cel puțin un x, astfel încît oricare ar fi y, avem $P(x, y)$“ și $(\exists y)\ (\forall x)\ P(x, y)$, adică „există cel puțin un y, astfel încît oricare ar fi x, avem $P(x, y)$“.

Este important să observăm că în cazul ultimelor categorii de propoziții (existențiale universalizate și universale existențializate) ordinea în care apar cuantorii este esențială.

Vom ilustra acest lucru printr-un exemplu. Fie τ mulțimea perechilor ordonate de numere reale și $P(x, y)$ predicatul „$x < y$“. Propoziția $(\forall x)\ (\exists y)P(x, y)$ adică „oricare ar fi x, există cel puțin un y, astfel încît $x < y$“ este adevărată (pentru orice număr real, fie acesta x_1, putem determina un număr real y_1, astfel încît $x_1 < y_1$; de exemplu, $y_1 = x_1 + 1$). În schimb propoziția $(\exists y)\ (\forall x)P(x, y)$, adică „există cel puțin un y, astfel încît oricare ar fi x, $x < y$“ este falsă (nu există un număr real care să fie mai mare decît toate numerele reale).

Exemplu:

Fie τ mulțimea numerelor naturale $\{5, 6, 7, 8\}$ și $P(x)$ predicatul: „$x < 7$“.
Propozițiile asociate predicatului sînt:
- propoziții particulare:
$P(5)$: „$5 < 7$“ cu valoarea de adevăr $\mathcal{A}$;
$P(6)$: „$6 < 7$“ cu valoarea de adevăr $\mathcal{A}$;
$P(7)$: „$7 < 7$“ cu valoarea de adevăr $\mathcal{F}$;
$P(8)$: „$8 < 7$“ cu valoarea de adevăr $\mathcal{F}$;
- propoziția universală:
$(\forall x)P(x)$: „oricare ar fi x, avem $x < 7$“
care este falsă (particularele $P(7)$ și $P(8)$ avînd valoarea $\mathcal{F}$);
- propoziția existențială:
$(\exists x)P(x)$: „există cel puțin un x, pentru care $x < 7$“, care este adevărată.

III.14.3.3. *Operații cu predicate*

Am văzut că o mulțime poate fi reprezentată ca mulțimea de adevăr a unui predicat. Fie acum două mulțimi α și β, determinate ca mulțimi de adevăr ale predicatelor $P(x)$ și $Q(x)$; τ este mulțimea totală din teoria mulțimilor:

$$\alpha = \{x \mid P(x)\} \text{ și } \beta = \{x \mid Q(x)\}.$$

Complementara mulțimii α este mulțimea $C\alpha$ formată din elemente din mulțimea totală, care nu aparțin lui α, adică pentru care proprietatea P este falsă:

$$C\alpha = \{x \mid \text{non } P(x)\}.$$

Intersecția mulțimilor α și β este mulțimea ale cărei elemente au și proprietatea P și proprietatea Q:

$$\alpha \cap \beta = \{x \mid P(x) \text{ și } Q(x)\}.$$

Reuniunea mulțimilor α și β este mulțimea elementelor pentru care cel puțin una din proprietăți P sau Q este adevărată:

$$\alpha \cup \beta = \{x \mid P(x) \text{ sau } Q(x)\}.$$

Diferența dintre mulțimea α și mulțimea β este mulțimea formată din elementele din α care nu sînt elemente ale lui β, adică:

$$\alpha - \beta = \{x \mid P(x) \text{ și non } Q(x)\}.$$

Mulțimea α este inclusă în β dacă orice element din α are proprietatea Q: $\alpha \subset \beta$ dacă și numai dacă „oricare ar fi x, dacă $P(x)$ atunci $Q(x)$“.

Mulțimea α este egală cu mulțimea β dacă orice element din α aparține lui β și dacă orice element din β aparține lui α: $\alpha = \beta$ dacă și numai dacă „oricare ar fi x, $P(x)$ dacă și numai dacă $Q(x)$“.

Observăm că pentru a determina mulțimile $C\alpha$, $\alpha \cap \beta$, $\alpha \cup \beta$, $\alpha - \beta$ am efectuat diferite operații logice cu predicate. Astfel, $C\alpha$ este mulțimea de adevăr a predicatului obținut prin negarea lui $P(x)$: $\neg P(x)$, $\alpha \cap \beta$ este mulțimea de adevăr a predicatului obținut prin conjuncția predicatelor care determină mulțimile α și β: $P(x) \wedge Q(x)$, $\alpha \cup \beta$ este determinată de $P(x) \vee Q(x)$ iar $\alpha - \beta$ de $P(x) \wedge \neg Q(x)$.

Relațiile de incluziune și egalitate între două mulțimi s-au definit cu ajutorul propozițiilor universale asociate implicației predicatelor $P(x)$ și $Q(x)$ $(\forall x)(P(x) \Rightarrow Q(x))$, respectiv echivalenței predicatelor $(\forall x)(P(x) \Leftrightarrow Q(x))$.

Fie două predicate $P(x)$ și $Q(x)$, cu mulțimea de referință τ și mulțimile de adevăr $\mathcal{D}_1$, respectiv $\mathcal{D}_2$.

Negația predicatului $P(x)$, $\neg P(x)$ are mulțimea de adevăr $\tau - \mathcal{D}_1$ (mulțimea elementelor care nu au proprietatea P). Conjuncția predicatelor $P(x)$ și $Q(x)$, $P(x) \wedge Q(x)$ are mulțimea de adevăr $\mathcal{D}_1 \cap \mathcal{D}_2$, iar disjuncția $P(x) \vee Q(x)$ are mulțimea de adevăr $\mathcal{D}_1 \cup \mathcal{D}_2$.

Implicația lui $Q(x)$ prin $P(x)$: „$P(x) \Rightarrow Q(x)$“ are mulțimea de adevăr $C\mathcal{D}_1 \cup \mathcal{D}_2$, iar echivalența predicatelor $P(x)$ și $Q(x)$ are mulțimea de adevăr $(C\mathcal{D}_1 \cup \mathcal{D}_2) \cap (C\mathcal{D}_2 \cup \mathcal{D}_1)$. Se spune că predicatele $P(x)$ și $Q(x)$ sînt echivalente dacă mulțimile lor de adevăr coincid ($\mathcal{D}_1 = \mathcal{D}_2$).

Cele afirmate anterior se pot urmări pe figurile III.6, unde zona haşurată reprezintă mulțimile de adevăr pentru diferitele predicate obținute din $P(x)$ și $Q(x)$ prin aplicarea operațiilor logice. Deci, așa cum în calculul propozițiilor construim formule logice cu propoziții simple sau compuse, la fel în calculul predicatelor vom putea construi formule cu predicate.

Cele cinci operații logice se pot efectua și cu propozițiile asociate predicatelor.

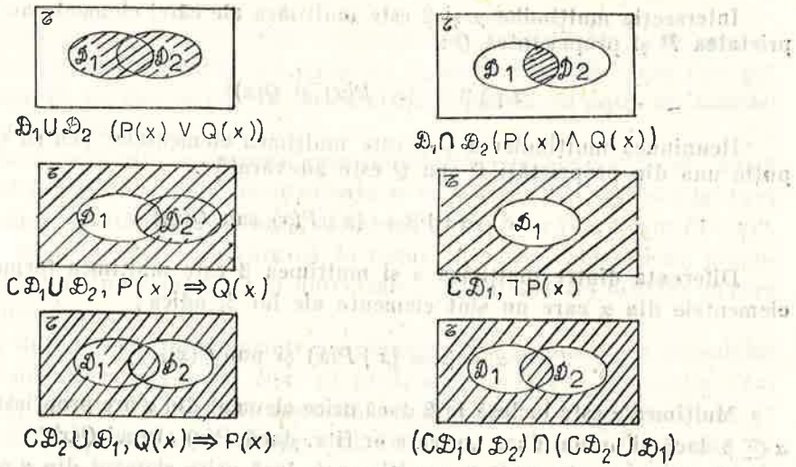


Fig. III.6

Fie predicatele $P(x)$ şi $Q(x)$. Fiecărei operaţii logice îi putem asocia diferite propoziţii :

a) *Propoziţii particulare*, de exemplu :

$\neg P(x_1)$; $\neg Q(x_1)$;
$P(x_1) \Rightarrow P(x_2)$; $Q(x_1) \Rightarrow Q(x_2)$;
$P(x_1) \Leftrightarrow P(x_2)$; $Q(x_1) \Leftrightarrow Q(x_2)$.

b) *Propoziţii universale* :

$(\forall x) \neg P(x)$;
$(\forall x)(P(x) \Rightarrow Q(x))$;
$(\forall x)(P(x) \Leftrightarrow Q(x))$.

c) *Propoziţii existenţiale* :

$(\exists x) \neg P(x)$;
$(\exists x)(P(x) \Rightarrow Q(x))$;
$(\exists x)(P(x) \Leftrightarrow Q(x))$.

Am utilizat ca operaţii logice numai $\neg$, $\Rightarrow$ şi $\Leftrightarrow$, deoarece ştim din calculul cu propoziţii că atît conjuncţia, cît şi disjuncţia se pot exprima cu ajutorul conectorilor $\neg$ şi $\Rightarrow$ (vezi exerciţiile 12, 13 de la pagina 89).

III.14.3.4. *Identităţi logice*

În calculul propoziţiilor am arătat importanţa pe care o au tautologiile, adică formulele care sînt întotdeauna adevărate, indiferent de valoarea de adevăr a propoziţiilor componente.

Se numeşte identitate logică o propoziţie asociată unui predicat, care are întotdeauna valoarea $\mathcal{A}$, indiferent de semnificaţia predicatului care a determinat-o.

Plecînd de la o anumită tautologie putem construi două categorii de identităţi logice :

a. Înlocuind toate propoziţiile P, Q, R, ... care apar, sau numai pe unele dintre ele, cu predicate. Astfel de exemple de identităţi logice sînt :

$P(x) \wedge Q(x) \Rightarrow P(x)$, din T.1 ;
$P(x) \Rightarrow P(x)$, din T.3 ;
$\neg \neg P(x) \Leftrightarrow P(x)$, din T.15 (negaţia negaţiei) ;
$P(x) \vee \neg P(x)$, din T.16 (principiul terţiului exclus) ;
$\neg (P(x) \wedge \neg P(x))$ din T.17 (principiul necontradicţiei) ;
$\neg (P(x) \wedge Q(x)) \Leftrightarrow \neg P(x) \vee \neg Q(x)$, din T.18 (De Morgan) ;
$P(x) \wedge Q \Rightarrow P(x)$, din T.1 ;
$(P(x) \Rightarrow Q) \wedge (Q \Rightarrow R(x)) \Rightarrow (P(x) \Rightarrow R(x))$, din T.21

(tranzitivitatea implicaţiei) ;
în ultimele două exemple Q fiind o propoziţie şi nu un predicat.

b. Construind propoziţiile asociate identităţilor obţinute prin procedeul enunţat anterior. De exemplu, din principiul identităţii $P \Rightarrow P$ putem obţine identităţile logice :

$P(x_1) \Rightarrow P(x_1)$, $\quad$ $P(x_1, y_1) \Rightarrow P(x_1, y_1)$;
$(\forall x)(P(x) \Rightarrow P(x))$, $\quad$ sau $\quad$ $(\forall x)(\forall y)(P(x, y) \Rightarrow P(x, y))$;
$(\exists x)(P(x) \Rightarrow P(x))$, $\quad$ $(\exists x)(\exists y)(P(x, y) \Rightarrow P(x, y))$.

Din negaţia negaţiei $\neg \neg P \Leftrightarrow P$ obţinem :

$\neg \neg P(x_1) \Leftrightarrow P(x_1)$, $\quad$ $\neg \neg P(x_1, y_1) \Leftrightarrow P(x_1, y_1)$
$(\forall x)(\neg \neg P(x) \Leftrightarrow P(x))$, $\quad$ sau $\quad$ $(\forall x)(\forall y)(\neg \neg P(x, y) \Leftrightarrow (P(x, y))$;
$(\exists x)(\neg \neg P(x) \Leftrightarrow P(x))$, $\quad$ $(\exists x)(\exists y)(\neg \neg P(x, y) \Leftrightarrow P(x, y))$.

Plecînd de la principiul terţiului exclus $P \vee \neg P$ avem :

$P(x_1) \vee \neg P(x_1)$, $\quad$ $P(x_1, y_1) \vee \neg P(x_1, y_1)$;
$(\forall x)(P(x) \vee \neg P(x))$, $\quad$ sau $\quad$ $(\forall x)(\forall y)(P(x, y) \vee \neg P(x, y))$;
$(\exists x)(P(x) \vee \neg P(x))$, $\quad$ $(\exists x)(\exists y)(P(x, y) \vee \neg P(x, y))$.

Plecînd de la definiţia cuantorilor $\forall$ şi $\exists$ putem obţine şi alte identităţi logice, care de fapt exprimă proprietăţi ale acestora.

Din definiţia cuantorului universal rezultă că dacă o propoziţie universală este adevărată, atunci este adevărată şi oricare dintre propoziţiile particulare ale acesteia. Obţinem astfel identitatea logică :

$$\text{IL.1} \quad (\forall x)P(x) \Rightarrow P(x_1)$$

sau, pentru un predicat binar :

$$(\forall x)(\forall y)P(x, y) \Rightarrow P(x_1, y_1).$$

● Analog, dacă o propoziţie particulară este adevărată, atunci este adevărată şi propoziţia existenţială, deci

$$\text{IL.2} \quad P(x_1) \Rightarrow (\exists x)P(x),$$

este o identitate logică.

În cazul unui predicat binar, IL.2 se scrie :

$$P(x_1, y_1) \Rightarrow (\exists x)(\exists y)P(x, y).$$

● Dacă o propoziție universală este adevărată, este evident că și propoziția existențială este adevărată și avem identitatea logică :

$$\text{IL.3 } (\forall x)P(x) \Rightarrow (\exists x)P(x)$$

sau

$$(\forall x)(\forall y)P(x, y) \Rightarrow (\exists x)(\exists y)P(x, y).$$

● *Negarea propozițiilor cuantificate*

Pentru ca propoziția $(\forall x)P(x)$ să nu fie adevărată este suficient ca să existe o valoare particulară x_1, pentru care $P(x_1)$ să fie falsă sau $\neg P(x_1)$ adevărată. Dar conform IL.2 :

$$\neg P(x_1) \Rightarrow (\exists x) \neg P(x).$$

Rezultă principiul negării unei propoziții universale :

$$\text{IL.4 } \neg((\forall x)P(x)) \Leftrightarrow (\exists x)\neg P(x),$$

sau, în cazul unui predicat binar :

$$\neg((\forall x)(\forall y)P(x, y)) \Leftrightarrow (\exists x)(\exists y)\neg P(x, y).$$

La fel, pentru ca $(\exists x)P(x)$ să fie falsă trebuie ca oricare ar fi x, $P(x)$ să fie falsă.

Negația unei propoziții existențiale se exprimă prin identitatea logică

$$\text{IL.5 } \neg((\exists x)P(x)) \Leftrightarrow (\forall x)\neg P(x)$$

sau, pentru predicate binare :

$$\neg((\exists x)(\exists y)P(x, y)) \Leftrightarrow (\forall x)(\forall y)\neg P(x, y).$$

Dar IL.3 și IL.4 ne conduc și la identitățile :

$$\text{IL.6 } (\forall x)\neg P(x) \Rightarrow \neg((\forall x)P(x)) \text{ respectiv}$$

$$(\forall x)(\forall y)\neg P(x, y) \Rightarrow \neg((\forall x)(\forall y)P(x, y)), \text{ iar IL.3 și IL.5 la}$$

$$\text{IL.7 } \neg((\exists x)P(x)) \Leftrightarrow (\exists x)\neg P(x) \text{ respectiv}$$

$$\neg((\exists x)(\exists y)P(x, y)) \Rightarrow (\exists x)(\exists y)\neg P(x, y).$$

Identitățile logice stau la baza construcției schemelor de raționamente din calculul predicatelor, pe care de altfel le-am întîlnit frecvent în rezolvarea diferitelor probleme de geometrie sau algebră.

Astfel, ori de cîte ori aplicăm, într-un caz particular, o teoremă, definiție sau axiomă, enunțată, pentru o clasă de obiecte, utilizăm IL.1, adică trecerea de la o propoziție universală la o propoziție particulară.

De exemplu, în geometria plană este adevărată propoziția : „oricare ar fi triunghiul ABC, dacă el este dreptunghic, atunci pătratul ipotenuzei este egal cu suma pătratelor catetelor".

Să presupunem că trebuie rezolvată problema :

Se dă un dreptunghi cu lungimea de 25 m și lățimea de 16 m (fig. III.7). Să se determine diagonala dreptunghiului.

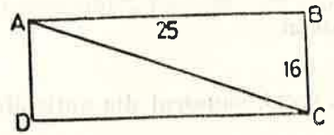


Fig. III.7

Din teorema lui Pitagora, prin IL.1 rezultă pentru ABC :

$$AC^2 = AB^2 + BC^2 = 25^2 + 16^2,$$

ceea ce ne permite să determinăm imediat diagonala dreptunghiului.

În continuare vom da un exemplu de rezolvare de probleme în care intervin calcule cu predicate.

O schemă de raționament specifică calculului cu predicate este *silogismul*. Silogismul are diferite forme, cea mai cunoscută fiind :

$$(\forall x)(P(x) \Rightarrow Q(x))$$

$$\frac{P(x_1)}{Q(x_1)}$$

În ipoteza că propozițiile $(\forall x)(P(x) \Rightarrow Q(x))$ și $P(x_1)$ sînt adevărate, rezultă adevărată propoziția particulară $Q(x_1)$.

Într-adevăr, dacă $(\forall x)(P(x) \Rightarrow Q(x))$ este adevărată, rezultă că oricare din propozițiile particulare asociate predicatului $P(x) \Rightarrow Q(x)$ este adevărată. Deci și $P(x_1) \Rightarrow Q(x_1)$ este adevărată. Dar $P(x_1)$ este adevărată. Rezultă din definiția implicației și $Q(x_1)$ adevărată.

Un exemplu foarte cunoscut de raționament prin silogism este următorul :

Toți oamenii sînt muritori
Socrate este om.

Deci Socrate este muritor.

Exemplu :

În algebră se demonstrează teorema : „pentru orice funcție de gradul doi $f(x) = ax^2 + bx + c$, $a \neq 0$, dacă $a > 0$ atunci f este strict descrescătoare pe intervalul $\left(-\infty, \frac{-b}{2a}\right]$ și strict crescătoare pe intervalul $\left[\frac{-b}{2a}, +\infty\right)$.

Fie $P(f)$: „$a > 0$" și $Q(f)$: „f strict descrescătoare pe intervalul $\left(-\infty, \frac{-b}{2a}\right]$ și strict crescătoare pe intervalul $\left[\frac{-b}{2a}, +\infty\right)$".

În mulțimea funcțiilor de gradul doi este adevărată propoziția: $(\forall f)(P(f) \Rightarrow Q(f))$.
Fie acum $f(x) = 2x^2 - x + 1$, deci $a = 2$.
$P(2x^2 - x + 1)$ este „$2 > 0$" și este adevărată.
Prin silogism rezultă $Q(f)$ adevărată, adică $2x^2 - x + 1$, strict descrescătoare în intervalul $\left(-\infty, \frac{+1}{4}\right]$ și strict crescătoare în intervalul $\left[\frac{1}{4}, +\infty\right)$.

Paradoxul mincinosului

Un paradox foarte vechi, cunoscut din antichitate, este paradoxul mincinosului.

Inițial avea forma unei întrebări P: Minți cînd spui că minți ?" adresată unui mincinos, adică unei persoane care nu spune niciodată adevărul. Mincinosul nu poate să răspundă decît mint" sau „nu mint".

Dacă răspunde „mint", el nespunînd niciodată adevărul, înseamnă că propoziția R: „mint" este falsă, deci $\neg R$ este adevărată, adică „nu mint" este adevărată, ceea ce contrazice ipoteza că el este un mincinos total.

Dacă răspunde „nu mint", analog se deduce că „mint" este adevărată, și mincinosul total spune un adevăr, ceea ce din nou contravine ipotezei.

Deci propoziției „mint" nu i se poate atribui nici valoarea $\mathcal{A}$, nici valoarea $\mathcal{F}$, ea luînd imediat valoarea contrară.

O altă formă a acestui paradox se numește „paradoxul lui Epimenide" și se enunță astfel: „Epimenide cretanul spunea că toți cretanii sînt mincinoși". Se ajungea aparent la aceeași contradicție încercînd să aflăm dacă ceea ce afirmă Epimenide este adevărat sau nu.

Dacă „toți cretanii sînt mincinoși", este adevărată, atunci Epimenide fiind cretan, el este mincinos și deci nu spune adevărul cînd afirmă că „toți cretanii sînt mincinoși", de unde contradicția cu ipoteza.

Dacă „toți cretanii sînt mincinoși" este falsă, înseamnă că Epimenide minte cînd face această afirmație, adică este de fapt adevărată propoziția „există cel puțin un cretan care nu minte". În această situație paradoxul dispare.

În general astfel de paradoxuri ascund o greșeală de logică. Astfe, să formalizăm paradoxul lui Epimenide. Fie $Q(x)$: „x este mincinos", atribuind cuvîntului mincinos sensul menționat anterior (un individ care nu spune niciodată adevărul), iar $P(x)$: „x este cretan".

Contradicția provenea din silogismul:

$$\frac{(\forall x)(P(x) \Rightarrow Q(x))}{\frac{P(\text{Epimenide})}{Q(\text{Epimenide})}}$$

Raționamentul este însă corect numai dacă premisele sînt adevărate, ceea ce nu putem afirma, deoarece nu știm dacă toți cretanii sînt mincinoși sau nu. Deci concluzia „Epimenide minte" nu poate fi afirmată.

Există însă multe alte variante ale acestui paradox. De exemplu:

a) Ce valoare de adevăr are: „Propoziția aceasta este falsă" ?

b) Într-o insulă trăiau niște uriași foarte răi și șireți. Orice străin care acosta pe insulă era jertfit fie zeului adevărului, fie zeului minciunii, în funcție de valoarea de adevăr a răspunsului pe care îl da străinul la o anumită întrebare. Unul dintre străini, fiind întrebat „Care îți va fi moartea ? "a răspuns „Voi fi jertfit zeului minciunii", răspuns care a pus în mare încurcătură pe uriași.

ÎNTREBĂRI ȘI EXERCIȚII

1. Ce se înțelege printr-un predicat ? Dați exemple de predicate unare și predicate binare.

2. Cîte tipuri de propoziții se pot asocia unui predicat ? Exemplificați în cazul predicatelor:

a) $P(x, y)$: „x divide pe y", cu τ mulțimea numerelor naturale;
b) $P(x)$: „$x^2 - 4x + 2 = 0$", cu τ mulțimea numerelor reale;
c) $P(x, y)$: „$(x \neq y) \wedge (x < y)$, cu τ mulțimea numerelor raționale.

3. Fie predicatele $P(x)$: „x are note mari la matematică" și $Q(x)$: „x are note mari la română", τ fiind mulțimea elevilor dintr-o clasă.

Să se scrie formalizat propozițiile:

a) Toți elevii au note mari la matematică, iar unii la română.
b) Unii elevi au note mici la română.
c) Fiecare elev are note mari la matematică și la română.
d) Nu toți elevii au note mici la matematică.
e) Există elevi care au note mari la matematică, dar mici la română.
f) Nici un elev nu are note mici la matematică și la română.
g) Dacă toți elevii au note mici la română, atunci cîțiva au note mari la matematică.
h) Dacă nu toți elevii au note mici la matematică, atunci unii nu au note mici nici la română.
i) Anca are note mari la matematică și Andrei are note mici la română.

4. Să se scrie formalizat și apoi în cuvinte negațiile propozițiilor din exercițiul 3.

5. Care este valoarea logică a propozițiilor din exercițiul 3 dacă:

— propoziția a) este adevărată;
— propoziția b) este adevărată;
— propoziția c) este adevărată.

6. Fie $P(x)$ şi $Q(x)$, două predicate cu aceeaşi mulţime de referinţă. Să se arate că următoarele propoziţii sînt identităţi logice :

a_1. $(P(x_1) \Rightarrow S) \Rightarrow ((\forall x)P(x) \Rightarrow S)$;
a_2. $((\exists x)P(x) \Rightarrow S) \Rightarrow (P(x_1) \Rightarrow S)$;
a_3. $(R \Rightarrow (\forall x)Q(x)) \Rightarrow (R \Rightarrow Q(x_1))$;
a_4. $(R \Rightarrow Q(x_1)) \Rightarrow (R \Rightarrow (\exists x)Q(x))$;
a_5. $((\forall x)P(x) \Rightarrow S) \Rightarrow ((\exists x)P(x) \Rightarrow S)$;
a_6. $(R \Rightarrow (\forall x)Q(x)) \Rightarrow (R \Rightarrow (\exists x)Q(x))$;
b_1. $(\forall x)(P(x) \Leftrightarrow Q(x)) \Rightarrow ((\forall x)P(x) \Leftrightarrow (\forall x)Q(x))$;
b_2. $(\forall x)(P(x) \Leftrightarrow S) \Rightarrow ((\forall x)P(x) \Leftrightarrow S)$;
b_3. $(\forall x)(P(x) \Leftrightarrow Q(x)) \Rightarrow ((\exists x)P(x) \Leftrightarrow (\exists x)Q(x))$;
b_4. $(\forall x)(P(x) \Leftrightarrow S) \Rightarrow ((\exists x)P(x) \Leftrightarrow S)$;
b_5. $S \Rightarrow (\forall x)(S \vee Q(x))$; $S \Rightarrow (\exists x)(S \vee Q(x))$;
c_1. $((\forall x)P(x) \vee (\forall x)Q(x)) \Rightarrow (\forall x)(P(x) \vee Q(x))$;
c_2. $(\exists x)(P(x) \vee Q(x)) \Leftrightarrow ((\exists x)P(x) \vee (\exists x)Q(x))$;
c_3. $(S \vee (\forall x)Q(x)) \Leftrightarrow (\forall x)(S \vee Q(x))$;
c_4. $(\exists x)(S \vee Q(x)) \Leftrightarrow S \vee (\exists x)Q(x)$;
c_5. $((\forall x)P(x) \vee (\forall x)Q(x)) \Rightarrow (\exists x)(P(x) \vee Q(x))$;
c_6. $(\forall x)(P(x) \vee Q(x)) \Rightarrow ((\exists x)P(x) \vee (\exists x)Q(x))$;
c_7. $(S \vee (\forall x)Q(x)) \Rightarrow (\exists x)(S \vee Q(x))$;
c_8. $(\forall x)(S \vee Q(x)) \Rightarrow (S \vee (\exists x)Q(x))$;
c_9. $(S \vee (\forall x)Q(x)) \Rightarrow (S \vee Q(x_1))$;
c_{10}. $(S \vee Q(x_1)) \Rightarrow (S \vee (\exists x)Q(x))$;
c_{11}. $(S \vee (\forall x)Q(x)) \Rightarrow (S \vee (\exists x)Q(x))$;
d_1. $(\forall x)(S \wedge Q(x)) \Rightarrow S$;
d_2. $(\exists x)(S \wedge Q(x)) \Rightarrow S$;
e_1. $((\forall x)P(x) \wedge (\forall x)Q(x)) \Leftrightarrow (\forall x)(P(x) \wedge Q(x))$;
e_2. $(\exists x)(P(x) \wedge Q(x)) \Rightarrow ((\exists x)P(x) \wedge (\exists x)Q(x))$;
e_3. $(S \wedge (\forall x)Q(x)) \Leftrightarrow (\forall x)(S \wedge Q(x))$;
e_4. $(S \wedge (\forall x)Q(x)) \Rightarrow (\exists x)(S \wedge Q(x))$;
e_5. $(S \wedge (\forall x)Q(x)) \Rightarrow (S \wedge (\exists x)Q(x))$;
e_6. $(S \wedge (\forall x)Q(x)) \Rightarrow (S \wedge (Q(x_1))$.

7. Definiţi relaţia binară :

a) simetrică ;
b) reflexivă ;
c) tranzitivă ;
d) antisimetrică.

8. Să se arate că în mulţimea numerelor reale relaţiile :

$$P(x, y) : \text{„}x = y\text{“}$$
$$R(x, y) : \text{„}|x - y| < 1\text{“}$$

sînt reflexive şi simetrice, iar relaţiile

$$S(x, y) : \text{„}x < y\text{“},$$
$$T(x, y) : \text{„}x - y = 1\text{“}$$

sînt reflexive şi antisimetrice.

De asemenea să se arate că relaţiile P şi S sînt tranzitive, iar T intranzitivă. R nu este nici tranzitivă, nici intranzitivă.

9. Daţi exemple de relaţii de ordine parţială şi de relaţii de echivalenţă.

10. Arătaţi că în mulţimea submulţimilor unei mulţimi γ „$\alpha \subset \beta$“ este o relaţie de ordine parţială.

Indicaţii şi răspunsuri

Pagina 67

3. Oricare. **4.** Se compară cifrele de acelaşi ordin. 1° $a = 9$, $b = 7$, $c \geqslant 1$ sau $a = 9$, $b > 7$; 2° $a < 9$ sau $a = 9$ şi $b < 7$. **5—8.** Se folosesc exemplele de la III.3.1 şi III.3.2. **9—14.** Analog cu **4**. **15—23.** Se folosesc exemplele de la III.4. **24—41.** Se folosesc exemplele de la III.4.2 şi III.4.3. **42—63.** Se folosesc exemplele de la III.11, III.12 şi III.13.

Pagina 88

2. a). b), c), d, f), h), i). **3.** Propoziţie simplă : c) ; propoziţii compuse : a), b), d), e). **9.** $P \vee Q \Leftrightarrow \neg(\neg P \wedge \neg Q)$. **10.** $P \wedge Q \Leftrightarrow \neg(\neg P \vee \neg Q)$. **11.** $P \Rightarrow Q \Leftrightarrow \neg(P \wedge \neg Q)$. **12.** $P \wedge Q \Leftrightarrow \neg(P \Rightarrow \neg Q)$. **13.** $P \vee Q \Leftrightarrow \neg P \Rightarrow Q$. **17.** Indicaţie. Se consideră următoarele propoziţii : P : „$ab = 0$“ ; Q : „$a = 0$“ ; $\neg Q$; R : „$b = 0$“ ; $\neg R$. Trebuie demonstrat $A : P \Rightarrow (Q \vee R)$; se presupune $\neg A$ şi se arată că $\neg A \Rightarrow \neg B$, cu B : „produsul a două numere nenule este nenul“. **18.** Nu se poate deduce care dintre persoane au spus adevărul.

Pagina 103

3. a) $(\forall x)P(x) \wedge \neg(\exists x)Q(x)$; b) $(\exists x)\neg Q(x)$; c) $(\forall x)(P(x) \wedge Q(x))$; d) $\neg(\forall x)\neg P(x)$; e) $(\exists x)(P(x) \wedge \neg Q(x))$; f) $\neg(\exists x)(\neg P(x) \wedge \neg Q(x))$; g) $(\forall x)\neg Q(x) \Rightarrow (\exists x)P(x)$; h) $\neg(\forall x)\neg P(x) \Rightarrow (\exists x)\neg\neg(Q(x)$; i) $P(\text{Anca}) \wedge \neg Q$ (Andrei). **4.** a) $(\exists x)\neg P(x) \vee (\forall x)\neg Q(x)$; b) $(\forall x)Q(x)$; c) $(\exists x)(\neg P(x) \vee \neg Q(x))$; d) $(\forall x)\neg P(x)$; e) $(\forall x)(\neg P(x) \vee Q(x))$; f) $(\exists x)(\neg P(x) \wedge \neg Q(x))$; g) $(\forall x)(\neg Q(x) \wedge \neg P(x))$; h) $(\exists x)P(x) \wedge (\forall x)\neg Q(x)$; i) $\neg P(\text{Anca}) \vee Q(\text{Andrei})$. **5.** a) $\mathcal{A}$; b) nedecis ; c) nedecis ; d) $\mathcal{A}$; e) nedecis ; f) $\mathcal{A}$; g) $\mathcal{A}$; h) $\mathcal{A}$; i) nedecis. Raţionament analog pentru celelalte două situaţii.

Bibliografie

1. BEZICOVICI S. I., *Calcule aproximative*, Editura Tehnică, București, 1952.
2. BOCȘA MINERVA, *Curs de analiză numerică*, Tip. Univ. Timișoara, 1971.
3. BRIȘCĂ V., IONESCU B., TUDOR GH., *Calcul numeric*, Editura Didactică și Pedagogică, București, 1973.
4. CÂMPEANU F., *De la papirusul Rhind la calculatorul electronic*, Editura Ion Creangă, București, 1976.
5. DEMIDOVITCH B., MARON I., *Elements de calcul numérique*, Editions MIR, Moscou, 1973.
6. DRĂGĂNESCU M., *Sistem și civilizație*, Editura Politică, București, 1976.
7. DUMITRIU A., *Soluția paradoxelor logico-matematice*, Editura Științifică, București, 1966.
8. GEORGESCU H., PREOTEASA P., *Introducere în sistemul de operare SIRIS*, Editura Albatros, București, 1978.
9. IONESCU S., BUCUR, *Asupra aproximării numerelor reale*, Gazeta Matematică, seria A, vol. LXXVI, nr. 12, București, 1971.
10. IONESCU H., *Calcule numerice*, Editura Tehnică, București, 1954.
11. KUNTZMANN J., *Méthodes numériques*, Ed. Hermann, Paris, 1969.
12. KRÎLOV N. A., *Lecții de calcule prin aproximații*, Editura Tehnică, București, 1957.
13. LAVROV I. A., MAXIMOVA L. L., *Probleme de teoria mulțimilor și logică matematică*, Editura Tehnică, București, 1974.
14. LIVOVSCHI L., *Bazele informaticii*, Editura Albatros, București, 1979.
15. MIHĂILESCU M., *Mijloace ajutătoare în calcule tehnice, Memorator matematic și tehnic*, Editura Tehnică, București, 1958.
16. MUNTEANU E., NEGRESCU L., PRODAN A., VUSCAN T., *Introducere în informatică*, Manual pentru cursurile postliceale și elevii liceelor pentru informatică, București, 1973.
17. MOISIL C. GR., *Elemente de logică matematică și teoria mulțimilor*, Editura Științifică, București, 1968.
18. NICULESCU ST., *Noțiuni de prelucrare automată a datelor*, Editura Didactică și Pedagogică, București, 1975.
19. NOVICOV P. S., *Elemente de logică matematică*, Editura Științifică, București, 1966.
20. POPOVICI P. C., *Calculatoare cu program și teoria programării*, Editura Științifică, București, 1972.
21. TEODORESCU N., *Calcul numeric și grafic*, Institutul Politehnic, București, 1951.
22. TODOR I., *Elemente de analiză numerică*, vol. I, curs litografiat de C.S.C.A.S.
23. TOMESCU I., *Articolele din Gazeta Matematică*, seria A, nr. 3–6, București, 1971.
24. VĂCARU V., VÎLCEANU S., *Calculatorul electronic — unealta secolului XX, Viitorul informaticii*, Editura Academiei R. S. România, București, 1970.
25. ALEF$_0$, ALGEBRA, *Numere reale, calcul numeric, numere complexe*, Editura Didactică și Pedagogică, București, 1974.

Coli de tipar : 6,75. B.T. 25.III.87
Apărut 1987

I. P. „Oltenia" Craiova
Str. Mihai Viteazul, nr. 4
Republica Socialistă România
Plan 33118 / 7 / 1987

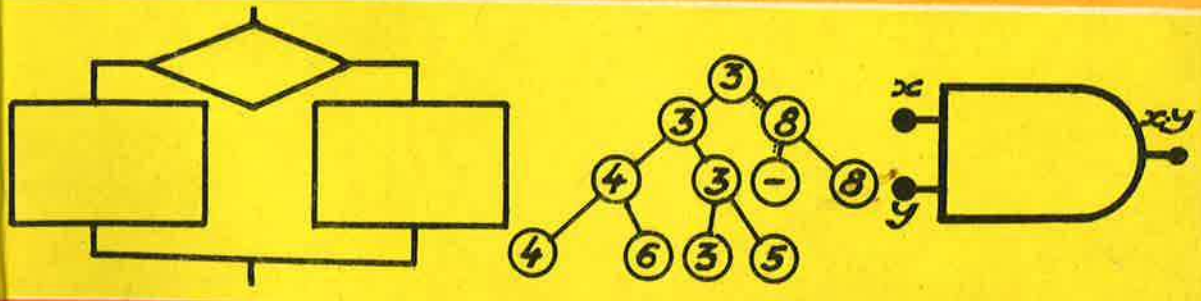
IOAN TOMESCU
ADRIAN LEU

MATEMATICĂ APLICATĂ ÎN TEHNICA DE CALCUL

X

clasa a X-a

Manual pentru licee cu profil
de matematică și de matematică-fizică



EDITURA DIDACTICĂ ȘI PEDAGOGICĂ
BUCUREȘTI – 1984

IOAN TOMESCU
ADRIAN LEU

MATEMATICĂ APLICATĂ ÎN TEHNICA DE CALCUL

X

Clasa a X-a

Manual pentru licee cu profil de matematică și de matematică-fizică

Editura Didactică și Pedagogică,
București

Manualul a fost elaborat pe baza programei şcolare aprobate de Ministerul Educaţiei şi Învăţămîntului cu nr. 39438/1980

Referenţi: *Conf. dr.* I. VĂDUVA
Cercetător ST. NICULESCU
Prof. I. V. MAFTEI

Redactor: *Prof.* ELEONORA DRĂGHIA
Tehnoredactor: VICTORIA GHIMIŞ
Coperta: N. SÎRBU

I. ALGEBRĂ BOOLEANĂ

I.1. Noţiunea de algebră booleană

I.1.1. **Latice**

În cadrul aritmeticii şi algebrei sînt studiate diferite operaţii cu numere În logica matematică s-au introdus operaţii ca: disjuncţia, conjuncţia sau negaţia propoziţiilor. În cazul mulţimii părţilor unei mulţimi s-au întîlnit operaţiile de reuniune, intersecţie şi complementară. Operaţiile amintite se deosebesc mult între ele, în primul rînd pentru că se efectuează asupra unor elemente aparţinînd unor mulţimi diferite.

Este cunoscut că operaţiile pot avea sau nu anumite proprietăţi. Operaţii diferite, care se efectuează asupra unor elemente distincte, pot avea aceleaşi proprietăţi. De exemplu, adunarea, înmulţirea, reuniunea, intersecţia, conjuncţia şi disjuncţia sînt operaţii comutative şi asociative.

Asemănarea unor operaţii din punctul de vedere al proprietăţilor pe care le posedă a dus la ideea de a nu se ţine seama de semnificaţia concretă a acestor operaţii sau a elementelor mulţimii pe care sînt definite, ci de a se lua în considerare numai proprietăţile lor comune. Trecerea de la operaţii determinate la operaţii nedeterminate este asemănătoare cu trecerea de la aritmetică la algebră, adică de la efectuarea unor operaţii cu numere cunoscute, la operaţii cu numere necunoscute.

Dacă pe o mulţime sînt definite una sau mai multe operaţii, care au anumite proprietăţi, vom spune că mulţimea a fost dotată cu o structură algebrică. Mulţimea însăşi, împreună cu operaţiile definite, se numeşte uneori algebră. Desigur, denumirea sugerează unele asemănări cu algebra obişnuită, dar trebuie subliniate cele două sensuri ale cuvîntului algebră şi anume ramură a matematicii, respectiv mulţime dotată cu o anumită structură.

Algebra booleană este un caz particular de algebră, operaţiile definite trebuind să satisfacă anumite relaţii bine determinate. Pentru a ajunge la definiţia structurii de algebră booleană se va prezenta mai întîi o structură mai simplă, denumită latice.

Definiţie. *Se numeşte* **latice** *o mulţime nevidă* M *înzestrată cu două operaţii binare, notate* $\sqcap$ *şi* $\sqcup$, *astfel încît pentru oricare elemente* $a, b, c \in M$ *să fie valabile următoarele proprietăţi:*

1. $a \sqcap b = b \sqcap a$ — 1′. $a \sqcup b = b \sqcup a$ (comutativitate)
2. $(a \sqcap b) \sqcap c = a \sqcap (b \sqcap c)$ — 2′. $(a \sqcup b) \sqcup c = a \sqcup (b \sqcup c)$ (asociativitate)
3. $a \sqcap (a \sqcup b) = a$ — 3′. $a \sqcup (a \sqcap b) = a$ (absorbţie)

O latice se notează $< M, \sqcap, \sqcup >$. Dacă nu există posibilitatea unei confuzii, notaţia poate fi simplificată, renunţînd la specificarea celor două operaţii.

Exemple

1. Mulțimea părților unei mulțimi M notată $\mathcal{P}(M)$, înzestrată cu operațiile de intersecție și reuniune formează o latice. Proprietățile 1, 2 și 3 pentru intersecție, respectiv 1', 2' și 3' pentru reuniune sînt cunoscute din teoria mulțimilor. Această latice poate fi notată $< \mathcal{P}(M), \cap, \cup >$.
2. Să considerăm mulțimea divizorilor pozitivi ai unui număr natural k. Această mulțime poate fi înzestrată cu două operații binare, care corespund calculului celui mai mare divizor comun și calculului celui mai mic multiplu comun. Astfel $a \sqcap b =$ cmmdc (a, b), iar $a \sqcup b =$ cmmmc (a, b). Proprietățile 1, 2, 1' și 2' sînt cunoscute, rezultînd direct din definițiile cmmdc și cmmmc. Proprietatea 3 se poate demonstra astfel: deoarece $a \sqcup b$ este cmmmc al numerelor a și b, rezultă că $a \sqcup b$ este un multiplu al numărului a; în acest caz $a \sqcap (a \sqcup b)$ adică cmmdc al numerelor a și $a \sqcup b$, nu poate fi decît a. Analog se demonstrează și proprietatea 3'. Această mulțime formează o latice, care poate fi notată $< V,$ cmmdc, cmmmc $>$, unde $V = \{x \mid x$ este un divizor pozitiv al lui $k\}$.

Observație. Analizînd cele șase proprietăți din definiția dată unei latice se remarcă unele similarități. Astfel, înlocuind în oricare dintre proprietăți o operație prin cealaltă (și reciproc) se obține tot o proprietate din definiție. Așadar, se poate enunța *principiul dualității* pentru latice, care are formularea:

Dacă într-o propoziție adevărată din teoria laticelor se înlocuiește o operație prin cealaltă (și reciproc) se obține de asemenea o propoziție adevărată.

Această a doua propoziție este numită *propoziția duală* corespunzătoare primei propoziții.

Principiul dualității permite să se evite demonstrarea unei propoziții atunci cînd propoziția sa duală este demonstrată.

Propoziție. *În orice latice L, pentru orice element $a \in L$ sînt adevărate relațiile* $a \sqcup a = a$ *și* $a \sqcap a = a$.

Demonstrație. Din proprietatea 3 rezultă $a = a \sqcap (a \sqcup b)$, astfel că

$$a \sqcup a = a \sqcup (a \sqcap (a \sqcup b)).$$

Notînd $a \sqcup b$ cu c, deoarece $c \in L$ rezultă conform proprietății 3' $a \sqcup (a \sqcap c) = a$, astfel că $a \sqcup a = a$.

Analog $a \sqcap a = a \sqcap (a \sqcup (a \sqcap b)) = a \sqcap (a \sqcup d) = a$, unde s-a notat $a \sqcap b$ cu d.

Observații

1. În orice latice se poate introduce o relație de ordine, notată $\sqsubseteq$, astfel: $a \sqsubseteq b \Leftrightarrow a \sqcup b = b$. Se verifică ușor că sînt adevărate cele trei proprietăți ale unei relații de ordine și anume:

   1) reflexivitate: $a \sqsubseteq a$, deoarece $a \sqcup a = a$;

   2) tranzitivitate: $a \sqsubseteq b$ și $b \sqsubseteq c \Rightarrow a \sqsubseteq c$, deoarece din $a \sqcup b = b$ și $b \sqcup c = c$ rezultă
   $$a \sqcup c = a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c = b \sqcup c = c;$$

   3) antisimetrie: $a \sqsubseteq b$ și $b \sqsubseteq a \Rightarrow a = b$, deoarece din $a \sqcup b = b$ și $b \sqcup a = a$ rezultă
   $$a = b \sqcup a = a \sqcup b = b.$$

2. Se poate arăta că relația de ordine poate fi definită și prin:
$$a \sqsubseteq b \Leftrightarrow a \sqcap b = a.$$

Într-adevăr, dacă $a \sqcup b = b$, atunci

$$a \sqcap b = a \sqcap (a \sqcup b) = a \text{ și, similar,}$$

dacă $a \sqcap b = a$, atunci

$$a \sqcup b = (a \sqcap b) \sqcup b = b \sqcup (b \sqcap a) = b.$$

În cazul celor două exemple de latice, relația de ordine corespunde relației de incluziune a două mulțimi și respectiv relației de divizibilitate a două numere.

Definiție. *Simbolul p se numește* **prim element** *în laticea L dacă pentru orice $x \in L$ este adevărată relația $p \sqsubseteq x$. Analog, u se numește* **ultim element** *în laticea L dacă pentru orice $x \in L$ este adevărată relația $x \sqsubseteq u$.*

Propoziție. *În orice latice finită există un prim element p și un ultim element u în raport cu relația de ordine $\sqsubseteq$.*

Demonstrație. Fie $L = \{a_1, a_2, \ldots, a_n\}$. Să arătăm că $p = a_1 \sqcap a_2 \sqcap \ldots \sqcap a_n$ (parantezele au fost omise datorită asociativității). Fie $x \in L$ un element oarecare al laticei L. Presupunînd că acest element se află în poziția i, putem considera $x = a_i$.

Astfel

$$p \sqcap x = (a_1 \sqcap a_2 \sqcap \ldots \sqcap a_i \sqcap \ldots \sqcap a_n) \sqcap a_i =$$
$$= a_1 \sqcap a_2 \sqcap \ldots \sqcap a_n = p,$$

adică $p \sqsubseteq x$.

Similar, $u = a_1 \sqcup a_2 \sqcup \ldots \sqcup a_n$, deoarece

$$u \sqcup x = (a_1 \sqcup a_2 \sqcup \ldots \sqcup a_i \sqcup \ldots \sqcup a_n) \sqcup a_i =$$
$$= a_1 \sqcup a_2 \sqcup \ldots \sqcup a_n = u,$$

adică $x \sqsubseteq u$.

Definiție. *Dacă într-o latice L, pentru oricare elemente $a, b, c \in L$ sînt satisfăcute relațiile*

$$a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c),$$

$$a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c),$$

laticea se numește **latice distributivă.**

Exemple

1. Laticea $< \mathcal{P}(M), \cap, \cup >$ este distributivă deoarece operațiile de reuniune și intersecție din teoria mulțimilor sînt distributive una față de cealaltă.

2. Laticea $< \{x \mid x$ este divizor pozitiv al lui $k\}$, cmmdc, cmmmc $>$ este distributivă deoarece pentru oricare trei numere $a, b, c \in \mathbf{N}$ sînt adevărate relațiile
   cmmmc $(a,$ cmmdc $(b, c)) =$ cmmdc (cmmmc $(a, b),$ cmmmc $(a, c))$ și
   cmmdc $(a,$ cmmmc $(b, c)) =$ cmmmc (cmmdc $(a, b),$ cmmdc $(a, c))$.

3. Fie mulțimea poligoanelor convexe din plan pentru care definim operațiile $\sqcap$ și $\sqcup$ astfel:
   a) $P_1 \sqcap P_2 = P_1 \cap P_2$, adică intersecția din teoria mulțimilor aplicată celor două mulțimi de puncte ale suprafețelor poligonale P_1 și P_2.

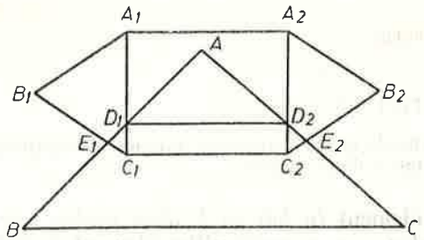


Fig. I.1

b) $P_1 \sqcup P_2$ este cel mai mic poligon convex care conține pe $P_1 \cup P_2$.

Aceste operații verifică relațiile din definiția unei latice (vezi ex. 6). Să arătăm că această latice nu este distributivă.

Fie $P = \triangle ABC$, $P_1 = \triangle A_1B_1C_1$ și $P_2 = \triangle A_2B_2C_2$ (vezi fig. I.1).

$$P_1 \sqcup P_2 = A_1B_1C_1C_2B_2A_2,$$

$$P \sqcap (P_1 \sqcup P_2) = AE_1C_1C_2E_2,$$

$$P \sqcap P_1 = \triangle C_1D_1E_1,$$

$$P \sqcap P_2 = \triangle C_2D_2E_2,$$

$$(P \sqcap P_1) \sqcup (P \sqcap P_2) = D_1E_1C_1C_2E_2D_2.$$

Se observă că $AE_1C_1C_2E_2 \neq D_1E_1C_1C_2E_2D_2$.

Observații

1. Cele două relații din definiția unei latice distributive sînt duale una în raport cu cealaltă, astfel că principiul dualității este valabil și în laticele distributive.
2. Se poate arăta că într-o latice cele două condiții de distributivitate sînt echivalente (vezi exercițiul 7).

I.1.2. Latice distributive și complementate

Definiție. *O latice distributivă* $< L, \sqcap, \sqcup >$ *se numește* **complementată** *dacă sînt îndeplinite următoarele condiții:*

1. *există un element neutru* u *pentru operația* $\sqcap$ *(denumit element universal), care satisface relația*

$$a \sqcap u = a \text{ pentru orice } a \in L;$$

2. *există un element neutru* n *pentru operația* $\sqcup$ *(denumit element nul), care satisface relația*

$$a \sqcup n = a \text{ pentru orice } a \in L;$$

3. *pentru orice element* $a \in L$ *există un element complementar* $\bar{a} \in L$, *care satisface relațiile*

$$a \sqcap \bar{a} = n \text{ și}$$

$$a \sqcup \bar{a} = u.$$

Exemple

1. Laticea $< \mathcal{P}(M), \cap, \cup >$ este o latice distributivă complementată, deoarece mulțimea M este element neutru pentru intersecție, mulțimea vidă $\emptyset$ este element neutru pentru reuniune, iar pentru orice mulțime $A \subseteq M$, mulțimea $\complement_M A = M - A$ este o mulțime complementară a mulțimii A.
2. Dacă numărul k este compus doar din factori primi la puterea 1, atunci laticea $< \{x \mid x$ este divizor pozitiv al lui $k\}$, cmmdc, cmmmc $>$ este distributivă și complementată.

Numărul k este element neutru pentru prima operație, deoarece cmmdc $(x, k) = x$ pentru orice x divizor pozitiv al lui k, iar numărul 1 este element neutru pentru a doua operație deoarece cmmmc $(x, 1) = x$ pentru orice x.

Dacă pentru un element x considerăm numărul $\frac{k}{x}$, atunci cmmdc $\left(x, \frac{k}{x}\right) = 1$ și cmmmc $\left(x, \frac{k}{x}\right) = k$, deoarece aceste numere nu au divizori comuni. Rezultă, deci, că numărul $\frac{k}{x}$ este complementar numărului x.

Dacă numărul k are factori primi la puteri mai mari decît 1, atunci laticea divizorilor nu este complementată. Considerînd $k = 60 = 2^2 \cdot 3 \cdot 5$, nu se poate găsi pentru numărul 10 un număr care să satisfacă condițiile de complementaritate. Prima condiție este satisfăcută de numerele 1 și 3 care nu satisfac însă și cea de-a doua condiție (cmmmc $(10, 1) = 10 \neq 60$ și cmmmc $(10, 3) = 30 \neq 60$).

Propoziție. *Într-o latice distributivă și complementată fiecare element are un singur element complementar.*

Demonstrație. Fie o latice distributivă și complementată L și să presupunem, prin reducere la absurd, că există $a \in L$, care are două elemente complementare, $\bar{a}$ și $\bar{a}'$, distincte $(\bar{a} \neq \bar{a}')$.

$\bar{a} = \bar{a} \sqcup n$ (n este element neutru pentru $\sqcup$)
$= \bar{a} \sqcup (a \sqcap \bar{a}')$ ($\bar{a}'$ este complement al lui a)
$= (\bar{a} \sqcup a) \sqcap (\bar{a} \sqcup \bar{a}')$ (distributivitatea operației $\sqcup$ față de $\sqcap$)
$= u \sqcap (\bar{a} \sqcup \bar{a}')$ ($\bar{a}$ este complement al lui a)
$= \bar{a} \sqcup \bar{a}'$ (u este element neutru pentru $\sqcap$).

Analog,

$\bar{a}' = \bar{a}' \sqcup n = \bar{a}' \sqcup (a \sqcap \bar{a}) = (\bar{a}' \sqcup a) \sqcap (\bar{a}' \sqcup \bar{a}) =$
$= u \sqcap (\bar{a}' \sqcup \bar{a}) = \bar{a}' \sqcup \bar{a}.$

Deci

$\bar{a} = \bar{a} \sqcup \bar{a}'$
$= \bar{a}' \sqcup \bar{a}$ (comutativitatea operației $\sqcup$)
$= \bar{a}',$

în contradicție cu ipoteza $\bar{a} \neq \bar{a}'$.

Definiție. *O latice distributivă și complementată se numește* **algebră booleană.**

O algebră booleană se va nota $< A, \sqcap, \sqcup, \bar{}, u, n >$.

Observații

1. Putem considera, în virtutea teoremei precedente, că într-o algebră booleană se poate defini o operație unară, care constă din luarea complementului unui element, astfel că într-o algebră booleană se poate spune că există trei operații $\sqcap$, $\sqcup$ și $\bar{}$. Într-o algebră booleană cele trei operații sînt notate, de obicei, „$\cdot$“, „$+$“ și „$^-$“, iar cele două elemente neutre sînt notate, respectiv, prin 1 și 0.
2. Condițiile din definiția algebrei booleene nu sînt independente. S-a arătat, de exemplu, că pentru o latice distributivă este suficientă numai una dintre cele două condiții din definiție. Se poate arăta că pentru a defini o algebră booleană este suficient ca pentru oricare elemente a, b, c să fie adevărate relațiile

$$ab = ba, \qquad a + b = b + a;$$

$$a(b + c) = ab + ac, \qquad a + bc = (a + b)(a + c);$$

$$a1 = a \qquad a + 0 = a;$$

$$a\bar{a} = 0, \qquad a + \bar{a} = 1.$$

Celelalte relații, adică

$(ab)c = a(bc)$, $(a + b) + c = a + (b + c)$;

$a(a + b) = a$, $a + ab = a$

putînd fi deduse din primele.

3. Elementul nul este prim element, iar elementul universal este ultim element (vezi exercițiul 8).
4. S-a arătat, de către Stone, că orice algebră booleană finită $< A, \cdot, +, \bar{}, 1, 0 >$ poate fi pusă în corespondență bijectivă cu mulțimea părților unei mulțimi finite M, care este de asemenea, o algebră booleană $< \mathcal{P}(M), \cap, \cup, \mathbf{C}_M, M, \emptyset >$. În acest fel rezultă că numărul de elemente dintr-o algebră booleană finită A este 2^n, unde n reprezintă numărul de elemente ale mulțimii finite M asociate algebrei booleene A.

În continuare, se vor prezenta două exemple de algebre booleene utilizate în tehnica de calcul.

Algebra B_2.

Algebra B_2 este algebra booleană formată din două elemente. Aceste elemente vor fi chiar elementele neutre ale celor două operații : 0 și 1. Putem considera că această algebră este cea mai simplă algebră booleană. Cele trei operații ale ei se definesc astfel :

$$0 + 0 = 0, \qquad 0 \cdot 0 = 0, \qquad \bar{0} = 1,$$
$$0 + 1 = 1, \qquad 0 \cdot 1 = 0, \qquad \bar{1} = 0.$$
$$1 + 0 = 1, \qquad 1 \cdot 0 = 0,$$
$$1 + 1 = 1, \qquad 1 \cdot 1 = 1,$$

Pentru a arăta că relațiile din definiția algebrei booleene sînt satisfăcute, se verifică egalitățile înlocuind elementele a, b, c cu 0, și 1, în toate combinațiile posibile. De exemplu, absorbția poate fi verificată ca în tabelul I.1.

a	b	$a + b$	$a(a + b)$	ab	$a + ab$
(1)	(2)	(3)	(4)	(5)	(6)
0	0	0	0	0	0
0	1	1	0	0	0
1	0	1	1	0	1
1	1	1	1	1	1

Tabelul I.1

Identitatea coloanelor (1), (4) și (6) demonstrează valabilitatea relațiilor $a = a(a + b) = a + ab$.

Se observă că operația $\cdot$ coincide cu înmulțirea aritmetică, însă operația $+$ este diferită de adunarea aritmetică, deoarece $1 + 1 = 1$. Găsim, de exemplu, $a + b = \max(a, b)$, pentru orice $a, b \in B_2$.

Această algebră a avut o deosebită importanță în evoluția calculatoarelor. O însemnată contribuție teoretică și practică în domeniul algebrelor booleene și al aplicațiilor lor a adus matematicianul român Grigore C. Moisil (1906—1973).

Fig. 1.2

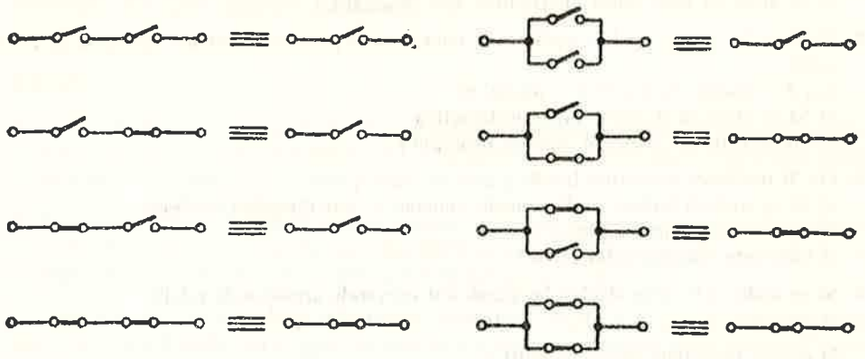


Fig. I.3

Elementele algebrei B_2 pot fi interpretate ca reprezentînd cei doi dipoli (rețea cu două borne) din figura I.2.

Operațiile $\cdot$ și $+$ sînt interpretate ca legarea dipolilor în serie, și respectiv, în paralel. Doi dipoli sînt considerați echivalenți dacă amîndoi, fie permit trecerea curentului electric, fie nu o permit.

Se poate observa, în figura I.3, că această interpretare este în concordanță cu definiția celor două operații.

Operația de complementare se interpretează analog și corespunde la schimbarea stării dipolului. Astfel, dacă dipolul permite trecerea curentului, dipolul complementar nu o va permite (și invers).

Algebra propozițiilor

În cadrul mulțimii propozițiilor se pot defini operațiile de disjuncție, conjuncție și negație. Dacă se consideră egale propozițiile echivalente, atunci relațiile de comutativitate, asociativitate și distributivitate corespunzătoare acestor operații sînt satisfăcute. Elementul universal este format de clasa tautologiilor (propoziții întotdeauna adevărate), iar elementul nul este reprezentat de clasa contradicțiilor (propoziții întotdeauna false). Se verifică ușor că aceste elemente neutre satisfac relațiile din definiția unei algebre booleene. Deci, mulțimea claselor de echivalență a propozițiilor formează o algebră booleană, fiecare clasă conținînd propoziții echivalente logic.

I.1.3. Exerciții

1. Să se arate că relația

$$x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap z$$

este verificată în orice latice dacă :

a) $x = z$,

b) $x = p$,

c) $z = u$.

2. O latice pentru care este verificată relația

$$x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap z$$

dacă $x \sqsubseteq z$, se numește latice modulară;
Să se arate că orice latice distributivă este modulară.

3. Fie $A = \{a_1, a_2, \ldots, a_n\}$, o mulțime de numere reale pe care se definesc operațiile $\sqcup$ și $\sqcap$ astfel:
$a \sqcup b = \max(a, b)$ și $a \sqcap b = \min(a, b)$.
a) Să se arate că A este o latice distributivă.
b) În ce condiții A este o algebră booleană?

4. Fie M mulțimea divizorilor lui 30.
a) Să se arate că laticea $< M$, cmmdc, cmmmc $>$ este o algebră booleană.
b) Cîte elemente are mulțimea M?
c) Care este complementul lui 6?

5. Să se arate că în orice algebră booleană sînt adevărate următoarele relații:
a) $aa = a$,
b) $\overline{\overline{a}} = a$, (principiul dublei negații)
c) $a0 = 0$,
d) $a + 1 = 1$,
e) $\overline{0} = 1$,
f) $\overline{1} = 0$,
g) $\overline{a + b} = \overline{a}\,\overline{b}$, (formulele lui De Morgan).
h) $\overline{ab} = \overline{a} + \overline{b}$.

6. Să se arate că mulțimea poligoanelor convexe din plan pentru care se definesc operațiile $\sqcap$ și $\sqcup$ astfel:
$P_1 \sqcap P_2 = P_1 \cap P_2$ (intersecția celor două poligoane),
$P_1 \sqcup P_2 =$ cel mai mic poligon convex care conține pe $P_1 \cup P_2$,
este o latice. Care sînt primul și ultimul element al acestei latice?

7. Să se arate că într-o latice L cele două condiții de distributivitate sînt echivalente.

8. Să se arate că într-o algebră booleană elementul nul n este prim element, iar elementul universal u este ultim element.

9. Să se efectueze următoarele calcule în algebra B_2:
a) $(1 \cdot 0 + 1 \cdot 1) \cdot (1 + 0 \cdot 1)$,
b) $((\overline{\overline{1 \cdot 0} + 1}) + 0) \cdot 1$,
c) $(0 + \overline{1} + \overline{0})(\overline{\overline{1} + 0})$.

1.2. Funcții booleene

1.2.1. **Expresii booleene**

Din aritmetică și algebră se știe că o expresie este un ansamblu de elemente (numere, litere) legate între ele prin simboluri care reprezintă operații matematice. Pentru a indica ordinea în care se efectuează operațiile, expresiile pot conține și paranteze.

În mod asemănător, se pot construi expresii cu elemente dintr-o algebră booleană, utilizînd simbolurile celor trei operații $\cdot$, $+$ și $^-$.

Definiție. *Se numește* **expresie booleană** *orice expresie rezultată prin aplicarea de un număr finit de ori a operațiilor* $+$, $\cdot$, $^-$ *unor elemente determinate sau nedeterminate ale unei algebre booleene.*

Exemple

1. $(a \cdot b) + \overline{(c \cdot 1)}$ este o expresie booleană care indică efectuarea operației „$\cdot$" între elementele a și b, a operației „$\cdot$" între elementele c și 1, a operației de complementare a rezultatului ultimei operații și a operației „$+$" între primul rezultat și ultimul.

2. Nu orice înșiruire de simboluri poate reprezenta o expresie. Astfel, $+a \cdot + b$, nu este o expresie booleană, deoarece nu respectă definiția. Simbolurile operațiilor care apar în acest șir nu leagă elementele în mod corect (există doi operatori consecutivi).

Ordinea în care trebuie efectuate operațiile dintr-o expresie este determinată de parantezele care apar în cadrul expresiei. Scrierea unei expresii poate fi simplificată prin omiterea unor paranteze dacă se introduc reguli de prioritate a operațiilor. De obicei, operația de complementare are prioritatea cea mai mare, fiind urmată de operația „$\cdot$", cea mai puțin prioritară fiind operația „$+$". La fel ca în algebra clasică, în expresiile booleene se poate omite simbolul operației „$\cdot$".
Astfel, expresia $(a \cdot b) + (c \cdot 1)$ se poate scrie $ab + c1$.

Definiție. *Fie o expresie booleană cu variabilele* $x_1, x_2, \ldots, x_n$. *Se numește* **valoare a expresiei booleene,** *pentru șirul de valori* $v_1, v_2, \ldots, v_n$, *valoarea obținută prin înlocuirea în expresie a variabilelor* $x_1, x_2, \ldots, x_n$ *cu valorile corespunzătoare* $v_1, v_2, \ldots, v_n$ *și efectuarea operațiilor indicate de simbolurile din expresie.*

Exemplu. Valoarea expresiei $x_1x_2 + \overline{x_2x_3} + \overline{x}_1x_2$ pentru șirul de valori 0, 1, 0 se obține astfel

$$0 \cdot 0 + \overline{1 \cdot 0} + \overline{0} \cdot 1 = 0 + \overline{0} + 1 \cdot 1 = 0 + 1 + 1 = 1.$$

Definiție. *Două expresii booleene cu aceleași variabile se numesc* **egale** *(sau echivalente) dacă pentru aceleași șiruri de valori ale variabilelor iau valori egale.*

Dacă numărul de elemente din algebra booleană este mic, se poate demonstra egalitatea a două expresii booleene prin verificarea egalității valorilor lor pentru toate șirurile de valori ale variabilelor.

Exemplu. În algebra booleană B_2 expresia $E_1(x, y) = \bar{x}\bar{y} + y$ este egală cu expresia $E_2(x, y) = x + y$, deoarece:

$$E_1(0, 0) = 0\bar{0} + 0 = 0 \cdot 1 + 0 = 0 + 0 = E_2(0, 0);$$

$$E_1(0, 1) = 0\bar{1} + 1 = 0 \cdot 0 + 1 = 0 + 1 = E_2(0, 1);$$

$$E_1(1, 0) = 1\bar{0} + 0 = 1 \cdot 1 + 0 = 1 + 0 = E_2(1, 0);$$

$$E_1(1, 1) = 1\bar{1} + 1 = 1 \cdot 0 + 1 = 0 + 1 = 1 + 1 = E_2(1, 1).$$

Egalitatea a două expresii booleene poate fi demonstrată și prin calcul boolean folosind proprietățile celor trei operații booleene.

Astfel, pentru expresiile E_1 și E_2 considerate anterior se poate demonstra egalitatea și prin:

$$\begin{aligned} E_1(x, y) &= x\bar{y} + y = \\ &= (x + y)(\bar{y} + y) = && \text{(comutativitate, distributivitate)} \\ &= (x + y)1 = && \text{(principiul terțului exclus)} \\ &= x + y = && \text{(1 este element neutru pentru „·“)} \\ &= E_2(x, y) \end{aligned}$$

Ca reguli de calcul boolean pot fi folosite atît relațiile din definiția algebrei booleene cît și alte relații stabilite pe baza acestora.

Exemple de calcul în algebra booleană.

1. $\overline{x + y + z}$

$= \overline{(x + y) + z}$ asociativitatea adunării

$= \overline{(x + y)}\,\bar{z}$ formula lui De Morgan

$= \bar{x}\bar{y}\bar{z}$ formula lui De Morgan

$= \bar{x}\bar{y}\bar{z}.$ asociativitatea înmulțirii.

2. $abc + a\bar{b}c + abc\bar{c} = ac(b + \bar{b}) + ab \cdot 0 = ac \cdot 1 + 0 = ac.$

1.2.2. Funcții booleene

În general, o funcție este definită ca o lege care asociază fiecărui element al unei mulțimi (numită domeniu de definiție) un singur element dintr-o altă mulțime (numită domeniu de valori). În concordanță cu mulțimile care alcătuiesc domeniul de definiție și domeniul de valori, funcțiile pot fi, de exemplu, funcții întregi de variabilă întreagă, funcții raționale de variabilă întreagă, funcții reale de variabilă reală. În cele ce urmează, se vor considera funcții la care atît domeniul de definiție cît și domeniul de valori sînt algebre booleene. Deoarece în practică algebra booleană cea mai răspîndită este algebra B_2, funcțiile booleene care vor fi studiate vor fi definite astfel:

Definiție. *Se numește* **funcție booleană** *(sau* **funcție binară***) de n variabile, o funcție definită pe produsul cartezian* $\underbrace{B_2 \times B_2 \times \ldots \times B_2}_{\text{de } n \text{ ori}}$ *cu valori în mulțimea* B_2.

Deoarece domeniul de definiție este un produs cartezian, un element al domeniului de definiție este un sistem ordonat alcătuit din n elemente din algebra B_2, adică din n elemente 0 sau 1. Pentru fiecare combinație de n elemente 0 sau 1, funcția îi asociază un element din algebra B_2 (adică 0 sau 1), numit *valoarea funcției* booleene pentru sistemul de n elemente considerat. O variabilă care ia valori în algebra B_2 se numește *variabilă booleană*. Putem astfel considera că funcțiile booleene sînt funcții de una sau mai multe variabile booleene.

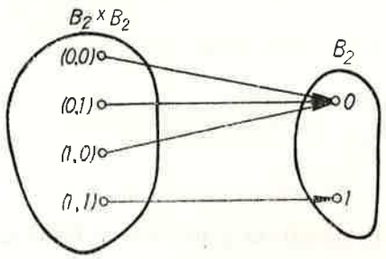


Fig. I.4

Exemplu. Putem defini o funcție $f: B_2 \times B_2 \to B_2$ astfel:

$f(0, 0) = 0,\ f(0, 1) = 0,\ f(1, 0) = 0,\ f(1, 1) = 1$

deoarece $B_2 \times B_2 = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. Această funcție poate fi reprezentată prin diagrama din figura I.4.

1.2.3. Reprezentarea funcțiilor booleene

Cea mai simplă formă de reprezentare a unei funcții booleene este tabelul său de valori. Domeniul de definiție al unei funcții booleene, B_2^n, are 2^n elemente astfel că tabelul va conține cele 2^n valori corespunzătoare ale funcției.

Exemple

1. Tabelul de valori pentru funcția din figura I.4 este

x_1x_2	00	01	10	11
f	0	0	0	1

Tabelul I.2

O altă formă a aceluiași tabel este următoarea:

x_1 \ x_2	0	1
0	0	0
1	0	1

Tabelul I.3

2. Pentru funcția din figura I.5 tabelul de valori poate fi:

$x_1x_2x_3$	000	001	010	011	100	101	110	111
g	1	0	0	0	1	0	1	1

Tabelul I.4

sau

x_1 \ x_2x_3	00	01	10	11
0	1	0	0	0
1	1	0	1	1

Tabelul I.5

3. Tabelul de valori poate fi alcătuit prin utilizarea unui cod ciclic pentru șirul de valori ale variabilelor:

x_1x_2	00	01	11	10
f	0	0	1	0

Tabelul I.6

Deoarece atît domeniul de definiţie al unei funcţii booleene, cît şi domeniul de valori sînt mulţimi finite, există un număr finit de funcţii booleene cu un domeniu de definiţie şi un domeniu de valori date. Astfel, mulţimea funcţiilor definite pe B_2^n cu valori în B_2 conţine 2^{2^n} funcţii. Aceste funcţii pot fi numerotate, asociind la fiecare funcţie un număr între 0 şi $2^{2^n} - 1$. O funcţie corespunde unui şir de 2^n valori 0 sau 1; aceste valori formează un număr binar cu 2^n cifre. Acest număr va fi denumit indexul funcţiei respective.

În tabelul I.7 sînt date toate funcţiile de o variabilă în număr de $2^{2^1} = 4$. Cele patru funcţii pot fi interpretate astfel:

f_0 este funcţia constantă 0,
f_1 este funcţia identică,
f_2 este funcţia complement,
f_3 este funcţia constantă 1.

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

Tabelul I.7

În tabelul I.8 sînt prezentate toate funcţiile de două variabile. Asupra celor 16 funcţii de două variabile se pot face mai multe observaţii. Astfel, funcţiile f_0 şi f_{15} sînt funcţiile constante 0 şi respectiv 1. Funcţiile f_3 şi f_5 sînt funcţiile identice cu x_1 şi respectiv x_2, iar f_{12} şi f_{10} sînt funcţiile identice cu complementul variabilei x_1 şi respectiv x_2. Funcţiile f_0 şi f_{15} nu depind de nici o variabilă, iar funcţiile f_3, f_5, f_{10} şi f_{12} nu depind decît de una din variabile. Celelalte funcţii depind de ambele variabile.

x_1x_2	00	01	10	11
f_0	0	0	0	0
f_1	0	0	0	1
f_2	0	0	1	0
f_3	0	0	1	1
f_4	0	1	0	0
f_5	0	1	0	1
f_6	0	1	1	0
f_7	0	1	1	1
f_8	1	0	0	0
f_9	1	0	0	1
f_{10}	1	0	1	0
f_{11}	1	0	1	1
f_{12}	1	1	0	0
f_{13}	1	1	0	1
f_{14}	1	1	1	0
f_{15}	1	1	1	1

Tabelul I.8

O altă formă de reprezentare a funcţiilor booleene utilizează expresiile booleene. O expresie booleană formată cu variabilele $x_1, x_2, \ldots x_n$ generează o funcţie booleană de n variabile booleene, care asociază fiecărui sistem de n elemente 0 sau 1 valoarea expresiei booleene pentru acest sistem de valori.

Se poate arăta că pentru orice funcţie booleană există o expresie booleană care o generează.

Exemple

1. Funcţia $f: B_2^2 \to B_2$ din figura I.4 poate fi generată de expresia x_1x_2. Aceasta se poate scrie:

$$f(x_1, x_2) = x_1x_2.$$

2. Funcţia $g: B_2^3 \to B_2$ din figura I.5 poate fi generată de expresia $x_1x_2 + \bar{x}_2\bar{x}_3$. Deci

$$g(x_1, x_2, x_3) = x_1x_2 + \bar{x}_2\bar{x}_3.$$

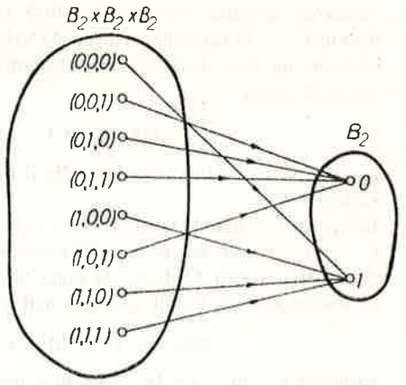


Fig. I.5

Să considerăm expresia $(x_1 + \bar{x}_2)(x_2 + \bar{x}_3)$. Această expresie generează o funcţie al cărei tabel de valori este:

$x_1x_2x_3$	$\bar{x}_2$	$x_1 + \bar{x}_2$	$\bar{x}_3$	$x_2 + \bar{x}_3$	$(x_1 + \bar{x}_2)(x_2 + \bar{x}_3)$
000	1	1	1	1	1
001	1	1	0	0	0
010	0	0	1	1	0
011	0	0	0	1	0
100	1	1	1	1	1
101	1	1	0	0	0
110	0	1	1	1	1
111	0	1	0	1	1

Tabelul I.9

Se observă că această funcţie coincide cu funcţia generată de expresia $x_1x_2 + \bar{x}_2\bar{x}_3$. Rezultă deci, că aceeaşi funcţie poate fi generată de mai multe expresii booleene.

Cele 16 funcţii de două variabile, prezentate în tabelul I.8, pot fi generate de următoarele expresii: $f_0 = 0$; $f_1 = x_1x_2$; $f_2 = x_1\bar{x}_2$; $f_3 = x_1$; $f_4 = \bar{x}_1x_2$; $f_5 = x_2$; $f_6 = \bar{x}_1x_2 + x_1\bar{x}_2$; $f_7 = x_1 + x_2$; $f_8 = \bar{x}_1\bar{x}_2$; $f_9 = \bar{x}_1\bar{x}_2 + x_1x_2$; $f_{10} = \bar{x}_2$; $f_{11} = x_1 + \bar{x}_2$; $f_{12} = \bar{x}_1$; $f_{13} = \bar{x}_1 + x_2$; $f_{14} = \bar{x}_1 + \bar{x}_2$; $f_{15} = 1$.

Funcţiile f_0, f_3, f_5, f_{10}, f_{12} şi f_{15} poartă denumirea de funcţii booleene degenerate de două variabile, deoarece valoarea lor nu depinde de ambele variabile. Celelalte funcţii poartă denumiri variate, care sînt provenite din teoria mulţimilor sau din calculul propoziţional. Astfel, se întîlnesc următoarele denumiri:

f_1 — intersecţie, conjuncţie, şi, produs;
f_2, f_4 — intersecţie indirectă;
f_6 — sumă disjunctivă, sau exclusiv, diferenţă simetrică;
f_7 — sumă logică, reuniune, disjuncţie, sau inclusiv;
f_8 — funcţia nici, funcţia lui Peirce;
f_9 — echivalenţă;
f_{11}, f_{13} — implicaţie;
f_{14} — excluziune, funcţia lui Sheffer.

Studierea funcțiilor booleene de două variabile are o importanță deosebită, deoarece se demonstrează că funcțiile booleene de trei sau mai multe variabile pot fi descompuse în funcții booleene de două variabile. De exemplu, funcția $g(x_1, x_2, x_3) = x_1x_2 + \bar{x}_2\bar{x}_3$ poate fi descompusă astfel

$$g(x_1, x_2, x_3) = f_7(f_1(x_1, x_2), f_8(x_2, x_3)).$$

Așadar, orice funcție booleană poate fi reprezentată cu ajutorul funcțiilor booleene de două variabile.

Dacă se ține seama de faptul că orice expresie booleană este formată utilizînd cele trei operații ($\cdot$, $+$ și $-$) și că aceste operații corespund funcțiilor f_1, f_7 și f_{12}, rezultă că sînt suficiente aceste trei funcții booleene de două variabile pentru a reprezenta orice funcție booleană. În aceste condiții, funcția g scrisă mai înainte poate fi descompusă astfel:

$$g(x_1, x_2, x_3) = f_7(f_1(x_1, x_2), f_1(f_{12}(x_2, 0), f_{12}(x_3, 0)).$$

Utilizînd formulele lui De Morgan se poate scrie că:

$$f_7(x_1, x_2) = f_{12}(f_1(f_{12}(x_1, 0), f_{12}(x_2, 0)), 0) \text{ și}$$

$$f_1(x_1, x_2) = f_{12}(f_7(f_{12}(x_1, 0), f_{12}(x_2, 0)), 0),$$

ceea ce permite să se afirme că orice funcție booleană poate fi reprezentată cu ajutorul a numai două funcții booleene de două variabile: fie f_1 și f_{12}, fie f_7 și f_{12}. Aceeași funcție g considerată anterior, se poate scrie de exemplu:

$$g(x_1, x_2, x_3) = f_{12}(f_1(f_{12}(f_1(x_1, x_2), 0), f_{12}(f_1(f_{12}(x_2, 0), f_{12}(x_3, 0)))), 0).$$

O proprietate remarcabilă o au însă funcțiile f_8 și f_{14}. Relațiile:

$f_1(x_1, x_2) = f_8(f_8(x_1, 0), f_8(x_2, 0))$;
$f_7(x_1, x_2) = f_8(f_8(x_1, x_2), 0)$;
$f_{12}(x_1) = f_8(x_1, 0)$

permit ca prin înlocuirea funcțiilor f_1, f_7 și f_{12}, cu expresiile indicate să se poată găsi pentru orice funcție booleană o reprezentare care să utilizeze numai funcția f_8 (funcția lui Peirce). Astfel, funcția g poate fi reprezentată sub forma:

$$g(x_1, x_2, x_3) = f_8(f_8(f_8(f_8(x_1, 0), f_8(x_2, 0)), f_8(x_2, x_3)), 0),$$

deoarece $g(x_1\ x_2, x_3) = f_7(f_1(x_1, x_2), f_8(x_2, x_3))$.

Funcția lui Peirce fiind o funcție binară este notată uneori ca o operație binară, utilizîndu-se simbolul $\downarrow$. Cu această notație putem reprezenta funcția g astfel:

$$g(x_1, x_2, x_3) = (((x_1 \downarrow 0) \downarrow (x_2 \downarrow 0)) \downarrow (x_2 \downarrow x_3)) \downarrow 0.$$

În mod analog se poate reprezenta orice funcție booleană utilizînd numai funcția lui Sheffer (vezi ex. 8).

Funcțiile lui Peirce și Sheffer nu au numai o importanță teoretică ci și una practică, deoarece circuitele electronice care realizează funcțiile lui Peirce și Sheffer sînt mai simple decît cele care realizează operațiile booleene $+$ și $\cdot$.

1.2.4. Exerciții

1. Să se restrîngă următoarele expresii:
   a) $(x + xy)(x + yz)$;
   b) $a(\bar{a} + b) + b(b + c) + \bar{b}$;
   c) $(a + b)(a + \bar{b})(\bar{a} + \bar{b})(\bar{a} + b)$;
   d) $xyz + x\bar{y}z + xy\bar{z} + x\bar{y}\bar{z}$.

2. Să se verifice următoarele identități:
   a) $a + \bar{a}b = a + b$;
   b) $(a + \bar{c})(b + c) = ac + b\bar{c}$;
   c) $ab + bc + ca = (a + b)(b + c)(c + a)$.

3. Fie expresiile:
   $E_1 = (a + ab)(a + b)$;
   $E_2 = \bar{a}\bar{b} + \overline{(a + b + c + d)}$;
   $E_3 = \overline{(a + b)(c + d)}$;
   $E_4 = a + b\bar{c} + \overline{\bar{a}\bar{b}\bar{c}}(ad + b)$.

   Să se arate că:

   a) $E_1 = a$; b) $E_2 = \bar{E}_4$; c) $E_3 = E_2 + \overline{\bar{c}\bar{d}}$; d) $E_4 = a + b$.

4. Să se calculeze valorile expresiilor:

   a) $(ab + c)(\bar{b} + c)$ pentru $a = 0$, $b = 1$ și $c = 1$;
   b) $ab\overline{(a + c)} + \overline{ab}$ pentru $a = 1$, $b = 1$ și $c = 0$;
   c) $\overline{\overline{ab + c} + \bar{a}\bar{c}}$ pentru $a = 0$, $b = 0$ și $c = 0$.

5. Să se verifice următoarele relații din algebra B_2 înlocuind nedeterminatele cu toate valorile posibile:
   a) $\bar{a}\bar{b} + a\bar{b} + ab = a + b$;
   b) $(a + b)(a + \bar{b}) = a$;
   c) $a\bar{c} + b + ac = a + b$;
   d) $ab + bc + \bar{a}c = ab + \bar{a}c$.

6. Să se afle care dintre următoarele expresii booleene sînt egale.
   a) $E_1 = ab + bc + ac$; $E_2 = \bar{a}bc + a\bar{b}c + ab$;
   b) $E_1 = \bar{a}\bar{b}\bar{c} + \bar{a}bd + abc + ab\bar{d}$; $E_2 = \overline{\bar{b}\bar{c}\bar{d}} + \bar{a}\bar{c}d + bcd + ac\bar{d}$;
   c) $E_1 = \bar{x}\bar{z} + \bar{x}y + x\bar{y}$; $E_2 = (\bar{x} + \bar{y})(x + y + \bar{z})$.

7. Să se arate că:
   a) $x\bar{z} + \bar{x}z = y$ dacă $x\bar{y} + \bar{x}y = z$;
   b) $ad + \bar{b}(\bar{c} + \bar{d}) = ad + ac + \bar{a}\bar{c} + \bar{b}c\bar{d}$ dacă $a\bar{d} + b\bar{c} = 0$.

8. Să se alcătuiască tabelul de valori al următoarelor funcții:
   a) $f(x_1, x_2) = x_1x_2 + x_2$;
   b) $g(x_1, x_2) = (x_1 + x_2)(\bar{x}_1 + x_1x_2)$;
   c) $f(x, y, z) = xz + \bar{y}\bar{z} + \bar{x}\bar{y}$;
   d) $g(x, y, z) = \bar{x}(y + \bar{z}) + \bar{y}(x + z)$.

9. Se consideră funcțiile:

   $g_1(a, b, c) = ab + bc + c\bar{a}$;
   $g_2(a, b, c) = (a + c)(b + \bar{a})$;

$g_3(a, b, c) = ab + c\bar{a}$;
$g_4(a, b, c) = (a + b)(b + c)(c + \bar{a})$;
$g_5(a, b, c) = ac + b\bar{a}$;
$g_6(a, b, c) = (a + b)(c + \bar{a})$.

Să se arate că:

a) funcțiile g_1, g_2 și g_3 sînt egale,
b) funcțiile g_4, g_5, g_6 sînt egale.

1.3. Forme canonice ale funcțiilor booleene

1.3.1. Forme normale ale funcțiilor booleene

În paragraful anterior s-a arătat că o aceeași funcție booleană poate fi reprezentată prin mai multe expresii booleene. Este de dorit să existe totuși o formă standardizată de reprezentare prin expresii a unei funcții booleene. Această formă de reprezentare trebuie să fie unică, pentru a permite determinarea rapidă a egalității a două funcții prin compararea expresiilor asociate.

Definiție. *Se numește* **produs elementar** *un produs de variabile booleene sau complemente ale acestora, fără ca aceeași variabilă să apară de mai multe ori.*

Exemplu. $a\bar{b}c$, $\bar{a}bc$, xyz, $\bar{x}_1\bar{x}_2$ sînt produse elementare iar $a + bc$, $\bar{x}y$ nu sînt produse elementare.

Definiție. *Se numește* **formă normală disjunctivă** *a unei expresii booleene o sumă de produse elementare egală cu expresia dată.*

Exemplu. $ab + bc + ac$, $\bar{x}\bar{y}\bar{z} + xyz$, $x_1\bar{x}_2$ sînt expresii sub formă normală disjunctivă.

Vom spune că o funcție booleană este scrisă sub forma normală disjunctivă dacă este reprezentată printr-o expresie sub forma normală disjunctivă.

Forma normală disjunctivă a unei funcții nu este unică, după cum se poate observa din exemplul următor

$$f(x, y, z) = xy + yz + \bar{x}z = xy + \bar{x}\bar{z} = xy + xyz + \bar{x}\bar{z}.$$

Putem astfel obține oricîte forme normale disjunctive utilizînd idempotența și absorbția.
Observație. Condiția ca într-un produs elementar să nu se repete variabilele are o justificare simplă. Dacă aceeași variabilă apare de mai multe ori, atunci aceasta poate fi redusă la o singură apariție prin idempotență. Dacă aceeași variabilă apare împreună cu complementul său atunci produsul respectiv este egal cu 0, astfel că poate fi eliminat din sumă.

În mod analog se definesc suma elementară și forma normală conjunctivă a unei funcții.

Definiție. *Se numește* **sumă elementară** *o sumă de variabile booleene sau complemente ale acestora, fără ca aceeași variabilă să apară de mai multe ori.*

De exemplu, $\bar{a} + \bar{b} + c$, $\bar{a} + b + c$, $x + y + z$, $\bar{x}_1 + \bar{x}_2$ sînt sume elementare iar $a + bc$, $\overline{x + y}$ nu sînt.

Definiție. *Se numește* **formă normală conjunctivă** *a unei expresii booleene un produs de sume elementare egal cu expresia dată.*

De exemplu $(a + b)(b + c)(a + c)$, $(\bar{x} + \bar{y} + \bar{z})(x + y + z)$, $x_1 + \bar{x}_2$ sînt expresii sub forma normală conjunctivă.

Nici forma normală conjunctivă a unei expresii nu este unică, după cum rezultă din exemplul următor:

$$(x + y)(y + z)(\bar{x} + z) = (x + y)(\bar{x} + z) = (x + y)(x + y + z)(\bar{x} + z).$$

Orice expresie booleană (deci și orice funcție) poate fi adusă la o formă normală disjunctivă și la o formă normală conjunctivă.

Pentru a aduce o expresie booleană la o formă normală disjunctivă se folosește următorul procedeu:

— dacă în cadrul expresiei operația de complementare este aplicată unor expresii, se aplică formulele lui De Morgan, pînă cînd în cadrul expresiei date nu mai apar decît complementele variabilelor;

— se distribuie operația „$\cdot$" în raport cu „$+$" ori de cîte ori este cazul;

— se elimină produsele care se anulează sau se repetă și variabilele care apar de două ori în același produs.

Exemplu. Expresia $E = \overline{xy}\cdot(x\bar{z} + \overline{x + \bar{z}}) + x\bar{y}\bar{z}$ poate fi transformată astfel:

$E = (\bar{x} + \bar{y})(x\bar{z} + \bar{x}z) + x\bar{y}\bar{z}$ (s-au aplicat formulele lui De Morgan)
$= \bar{x}xz + \bar{x}\bar{x}z + x\bar{y}z + \bar{x}\bar{y}z + x\bar{y}\bar{z}$ (s-a aplicat distributivitatea operației $\cdot$ față de $+$)
$= \bar{x}z + x\bar{y}z + \bar{x}\bar{y}z$ (s-au eliminat primul și ultimul produs).

S-a obținut pentru E o formă normală disjunctivă.

Dacă utilizăm și proprietatea de absorbție, putem elimina și ultimul produs, obținînd pentru expresia dată o altă formă normală:

$$E = \bar{x}z + x\bar{y}\bar{z}.$$

Pentru a aduce o expresie booleeană la o formă normală conjunctivă se folosește un procedeu similar, în care, după aplicarea formulelor lui De Morgan, se distribuie operația „$+$" în raport cu „$\cdot$" și se elimină sumele care au valoarea constantă 1 sau se repetă, precum și variabilele care apar de două ori în aceeași sumă.

Exemplu. Aceeași expresie E poate fi adusă la o formă normală conjunctivă astfel:

$$E = \overline{xy}(x\bar{z} + \overline{x + \bar{z}}) + x\bar{y}\bar{z} = (\bar{x} + \bar{y})(x\bar{z} + \bar{x}z) + x\bar{y}\bar{z} =$$
$$= (\bar{x} + \bar{y})(x + z)(\bar{x} + \bar{z}) + x\bar{y}\bar{z} =$$
$$= (\bar{x} + \bar{y} + x)(\bar{x} + \bar{y} + \bar{y})(\bar{x} + \bar{y} + \bar{z})(x + z + x)(x + z + \bar{y}) =$$
$$= (x + z + \bar{z})(\bar{x} + \bar{z} + x)(\bar{x} + \bar{z} + \bar{y})(\bar{x} + \bar{z} + \bar{z}) =$$
$$= (\bar{x} + \bar{y})(\bar{x} + \bar{y} + \bar{z})(x + z)(x + \bar{y} + z)(\bar{x} + \bar{z}).$$

Expresia se poate simplifica prin absorbție astfel că

$$E = (\bar{x} + \bar{y})(x + z)(\bar{x} + \bar{z}).$$

1.3.2. Forme canonice ale funcțiilor booleene

Definiție. *Se numește* **mintermen** *în raport cu variabilele booleene $x_1, x_2, \ldots, x_n$ un produs elementar în care apar, fie simple, fie complementate, toate variabilele $x_1, x_2, \ldots, x_n$.*

Textele prevăzute cu o bară laterală sînt destinate numai profilului de matematică.

Un mintermen are deci n factori. Acești factori se scriu în ordinea naturală a variabilelor care îi alcătuiesc. De exemplu,

$$x_1x_2\bar{x}_3,\ a\bar{b}\bar{c},\ \bar{x}\bar{y}\bar{z},\ ab,\ x_1x_2x_3x_4x_5.$$

Fiecărui mintermen i se poate asocia un număr binar astfel:

Variabilele complementate se înlocuiesc cu cifra 0, iar variabilele necomplementate cu cifra 1. Șirul de cifre obținut reprezintă un număr binar (în baza 2), care este asociat mintermenului respectiv. De exemplu, mintermenilor scriși anterior li se asociază numerele:

$(110)_2$, $(100)_2$, (000), $(11)_2$, $(11111)_2$, care reprezintă în baza zece numerele 6, 4, 0, 3, 31.

Considerînd n variabile booleene, numărul de mintermeni diferiți care se pot construi cu aceste variabile este egal cu numărul de numere binare cu n cifre. Acest număr este 2^n.

Mintermenii joacă un rol important în cadrul formelor de reprezentare a funcțiilor booleene, datorită următoarei proprietăți:

Un mintermen are valoarea 1 pentru un singur șir de valori ale variabilelor booleene care îl alcătuiesc.

Într-adevăr, deoarece un produs boolean are valoarea 1 dacă și numai dacă toți factorii săi au valoarea 1, iar factorii unui mintermen sînt variabile booleene simple sau complementate, alegînd valoarea 0 pentru variabilele care apar complementate în cadrul mintermenului și valoarea 1 pentru celelalte variabile, se obține combinația de valori căutată.

Se definește în mod analog noțiunea de maxtermen.

Definiție. *Se numește* **maxtermen** *în raport cu variabilele booleene* $x_1, x_2, \ldots, x_n$ *o sumă elementară în care apar, fie simple, fie complementate, toate variabilele* $x_1, \ldots, x_n$.

Un maxtermen are deci n termeni, care se scriu în ordinea naturală a variabilelor care îi alcătuiesc. De exemplu

$$x_1 + \bar{x}_2 + x_3,\ \bar{x} + \bar{y} + z,\ \bar{a} + \bar{b} + \bar{c},\ x + y,\ x_1 + x_2 + x_3 + x_4 + x_5.$$

Pentru un maxtermen se pot face aceleași observații ca pentru mintermeni. Asocierea unui număr binar pentru fiecare maxtermen se face înlocuind variabilele complementate cu cifra 1, iar variabilele necomplementate cu cifra 0. Numărul de maxtermeni care se pot forma cu n variabile booleene este, de asemenea, 2^n.

Maxtermenii au o proprietate similară cu aceea a mintermenilor, și anume:

Un maxtermen are valoarea 0 pentru o singură combinație de valori ale variabilelor booleene, care îl alcătuiesc (0 pentru variabilele simple și 1 pentru variabilele complementate).

Mintermenii permit alcătuirea unor forme normale particulare, deosebit de importante.

Definiție. *Se numește* **formă canonică disjunctivă** *a unei expresii booleene cu n variabile o formă normală disjunctivă echivalentă cu această expresie, alcătuită numai din mintermeni cu n variabile.*

Forma canonică disjunctivă a unei expresii se mai numește și *formă normală disjunctivă perfectă.*

Exemple. $a\bar{b}c + \bar{a}bc + \bar{a}\bar{b}c$, $xyz + \bar{x}yz + x\bar{y}z$, $x_1x_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2x_3x_4 + x_1x_2x_3x_4$ sînt forme canonice disjunctive.

În aceste exemple, primele 2 expresii au cîte 3 variabile, iar ultima expresie are 4 variabile.

Propoziție. *Forma canonică disjunctivă a unei funcții booleene este unică, dacă facem abstracție de ordinea mintermenilor.*

Demonstrație. Să presupunem că pentru o funcție booleană există două forme canonice disjunctive distincte.

Deci există un mintermen care face parte dintr-una din formele canonice (să spunem din prima formă) și nu face parte din cealaltă. Să considerăm combinația de valori ale variabilelor pentru care mintermenul respectiv are valoarea 1 și să analizăm valorile celor două expresii corespunzătoare acestei combinații. Deoarece pentru această combinație de valori ale variabilelor mintermenul din prima expresie are valoarea 1, indiferent de valorile celorlalți mintermeni, prima expresie are valoarea 1. Pentru a doua expresie se obține însă valoarea 0, deoarece toți mintermenii acestei expresii nu pot avea decît valoarea 0 (conform proprietății mintermenilor prezentată anterior). Rezultă, deci, că aceste două expresii nu sînt echivalente, adică presupunerea făcută este falsă.

Cele două forme canonice conțin aceeași mintermeni, ceea ce încheie demonstrația.

Forma canonică disjunctivă poate fi deci considerată ca o reprezentare standard a funcțiilor booleene.

Totuși, există o singură funcție booleană care nu poate fi reprezentată sub forma canonică disjunctivă și anume funcția constantă 0. Aceasta este o consecință a faptului că fiecărui mintermen dintr-o formă canonică disjunctivă îi corespunde o combinație de valori ale variabilelor pentru care funcția respectivă ia valoarea 1, astfel că o funcție reprezentată sub forma canonică disjunctivă ia cel puțin o dată valoarea 1.

Forma canonică disjunctivă a unei funcții booleene poate fi obținută atît pe baza tabelei de valori a funcției, cît și pe baza unei forme normale disjunctive a acesteia.

Tabela de valori a unei funcții booleene permite obținerea formei canonice disjunctive a funcției astfel:

Pentru fiecare combinație de valori ale variabilelor booleene pentru care funcția are valoarea 1 se consideră mintermenul corespunzător (adică acel mintermen care ia valoarea 1 numai pentru această combinație). Suma tuturor acestor mintermeni constituie forma canonică disjunctivă a funcției. Reamintim că mintermenul corespunzător unei combinații de valori ale variabilelor este produsul variabilelor care iau valoarea 1 și a complementelor variabilelor care iau valoarea 0.

Fie, de exemplu, funcția definită de următoarea tabelă de valori

a	0	0	0	0	1	1	1	1
b	0	0	1	1	0	0	1	1
c	0	1	0	1	0	1	0	1
f	1	0	0	0	1	0	1	1

Tabelul I.10

Funcția ia de 4 ori valoarea 1, pentru următoarele combinații de valori ale variabilelor a, b, c: (0, 0, 0), (1, 0, 0), (1, 1, 0), (1, 1, 1).

Acestor combinații de valori le corespund următorii mintermeni: $\bar{a}\bar{b}\bar{c}$, $a\bar{b}\bar{c}$, $ab\bar{c}$, abc. Forma canonică disjunctivă a acestei funcții este deci:

$$f(a, b, c) = \bar{a}\bar{b}\bar{c} + a\bar{b}\bar{c} + ab\bar{c} + abc.$$

O formă normală disjunctivă a unei funcții poate fi transformată într-o formă canonică disjunctivă astfel:

Fiecare termen al formei normale disjunctive care nu este mintermen, adică fiecare termen care nu conține toate variabilele, se înmulțește cu expresii de forma $(x + \bar{x})$, pentru fiecare variabilă x absentă din acel termen. Ținînd cont de distributivitatea înmulțirii booleene față de adunarea booleană se desfac parantezele și se elimină termenii dubli (ținînd seama de idempotența adunării booleene). Se poate observa ușor că expresia obținută nu conține decît mintermeni și este echivalentă cu expresia inițială. Rezultă deci că s-a obținut forma canonică a funcției respective.

Exemplu. Fie forma normală disjunctivă:

$$f(a, b, c) = b\bar{c} + a\bar{c} + ab.$$

Aceasta se transformă astfel:

$$b\bar{c} + a\bar{c} + ab = b\bar{c}(a + \bar{a}) + ac(b + \bar{b}) + ab(c + \bar{c}) =$$

$$= ab\bar{c} + \bar{a}b\bar{c} + abc + a\bar{b}c + abc + abc =$$

$$= ab\bar{c} + \bar{a}b\bar{c} + abc + abc.$$

În mod analog, utilizînd noțiunea de maxtermen, se definește forma canonică conjunctivă a unei expresii booleene. Forma canonică conjunctivă permite de asemenea o reprezentare standard a funcțiilor booleene deoarece și această formă are proprietatea de unicitate. Singura funcție care nu poate fi reprezentată sub formă canonică conjunctivă este funcția constantă 1.

De asemenea, forma canonică conjunctivă se poate obține pe baza fie a tabelei de valori a funcției, fie a unei forme normale conjunctive a acesteia.

1.3.3. **Exerciții**

1. Să se găsească o formă normală disjunctivă a următoarelor expresii:

   a) $(x + xy)(x + y)$;

   b) $(\overline{x_1x_2})(\overline{x_1 + x_2})(\overline{x_3 + x_4})$;

   c) $\overline{(a + b)(c + d)}$;

   d) $(a + b)\bar{c} + \bar{a}bc(ad + b)$.

2. Să se găsească o formă normală conjunctivă a expresiilor de la exercițiul 1.

3. Să se indice care dintre următoarele forme normale sînt și canonice.

   a) $f(x, y, z) = x\bar{y} + y\bar{z}$;

   b) $f(a, b, c) = ab\bar{c} + \bar{a}\bar{b}\bar{c} + a\bar{b}c + \bar{a}bc$;

   c) $f(x_1, x_2, x_3, x_4) = x_1x_2x_3x_4 + \bar{x}_1\bar{x}_2x_3\bar{x}_4 + x_1\bar{x}_2x_3x_4$;

   d) $g(x, y) = x + \bar{y}$;

   e) $g(x, y) = xy$;

   f) $f(a, b) = (a + b)(\bar{a} + \bar{b})$;

   g) $g(x, y, z) = (\bar{x} + \bar{y} + \bar{z})(y + z)(x + y + z)$;

   h) $f(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3 + x_3\bar{x}_4)(\bar{x}_1 + x_2 + \bar{x}_3 + x_4)$.

4. Să se afle forma canonică disjunctivă a următoarelor expresii:

   a) $ab + \bar{b}\bar{c}$;

   b) $\bar{x}\bar{y}z + yz$;

   c) $\bar{x}_2x_3 + \bar{x}_1x_3$.

5. Să se găsească forma canonică conjunctivă a expresiilor de la exercițiul 4.

6. Fie funcția $f(a, b, c, d)$ definită de tabelul I.11.
   a) Să se scrie forma canonică disjunctivă a acestei funcții.
   b) Să se scrie forma canonică conjunctivă a funcției f.
   c) Să se verifice prin calcul boolean că expresiile obținute la punctele a) și b) sînt echivalente.

a	b	c	d	$f(a, b, c, d)$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Tabelul I. 11

1.4. Simplificarea funcțiilor booleene

1.4.1. **Simplificarea prin calcul boolean**

O problemă importantă legată de reprezentarea funcțiilor booleene o constituie găsirea unei forme cît mai simple pentru expresiile acestora.

Simplificarea prin calcul boolean este asemănătoare cu simplificarea expresiilor algebrice obișnuite. Trebuie însă să se țină cont de regulile specifice ale calculului boolean.

În acest sens sînt deosebit de utile regulile de idempotență ($a + a = a$ și $aa = a$), de absorbție ($a + ab = a$, $a(a + b) = a$) și de combinare ($ab + a\bar{b} = a$, $(a + b)(a + \bar{b}) = a$).

Aceste reguli pot fi folosite atît ca reguli de simplificare, cît și ca artificii de calcul pentru introducerea unor noi termeni necesari pentru combinațiile ulterioare.

Exemple

1. $abcd + ab\bar{c} + abc + abd + \bar{a}bcd + \bar{a}bd =$

   $= ab\bar{c} + abc + abd + \bar{a}bd =$ (prin absorbție)

   $= ab + bd$ (prin combinare).

2. $ab + \bar{a}bc + a\bar{b}\bar{c} =$

   $= ab + abc + \bar{a}bc + a\bar{b}\bar{c} =$ (prin absorbție) (artificiu de calcul)

   $= ab + abc + \bar{a}bc + abc + a\bar{b}\bar{c} =$ (prin idempotență) (artificiu de calcul)

   $= ab + (a + \bar{a})bc + ac(b + \bar{b}) =$

   $= ab + bc + ac$ (prin combinare).

Totuși calculul boolean direct nu asigură obținerea celei mai simple expresii. Aflarea celei mai simple expresii trebuie să țină seama de toate combinațiile posibile, efectuînd o căutare metodică. Trebuie stabilit, însă, ce se înțelege prin expresie mai simplă.

O expresie va fi considerată mai simplă decît alta dacă numărul de apariții ale variabilelor din prima expresie este mai mic decît numărul corespunzător din cealaltă.

Astfel expresia $ab + \bar{a}c$ (în care variabilele apar de patru ori) va fi considerată mai simplă decît $a + b + \bar{a}\bar{b}\bar{c}$ în care variabilele apar de cinci ori.

De obicei se caută cea mai simplă expresie în formă normală disjunctivă, echivalentă cu expresia dată.

I.4.2. **Simplificarea prin diagrame**

Utilizarea diagramelor pentru simplificarea funcțiilor booleene este o metodă utilizată de obicei pentru funcțiile reprezentate prin tabele.

Diagramele Euler-Venn sînt cunoscute din teoria mulțimilor. S-a arătat însă că $< \mathcal{P}(M), \cap, \cup >$ este o algebră booleană, astfel că aceste diagrame pot fi utilizate și la reprezentarea geometrică a relațiilor din algebra booleană B_2. Într-o diagramă Euler-Venn fiecare variabilă a expresiei booleene se reprezintă printr-o mulțime, de obicei un cerc. Interiorul mulțimii corespunde variabilei respective, iar exteriorul mulțimii (deci complementara mulțimii) corespunde complementului variabilei. Un termen al unei expresii booleene corespunde intersecției mulțimilor corespunzătoare variabilelor care îl compun. În cazul unei variabile care apare sub formă de complement se consideră desigur intersecția cu exteriorul mulțimii corespunzătoare.

Intreaga expresie va corespunde astfel reuniunii mulțimilor corespunzătoare fiecărui termen.

În figurile I.6, I.7 și I.8 sînt prezentate cîteva exemple de diagrame Euler-Venn, corespunzînd termenilor a, $a\bar{b}$ și, respectiv, $a\bar{b}c$. Se poate observa că unui mintermen îi corespunde o zonă care nu mai este divizată în alte zone. În figura I.9 au fost înscriși mintermenii corespunzători fiecărei zone elementare. Considerînd expresia

$$a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c} + abc,$$

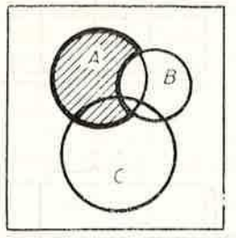


Fig. I.6

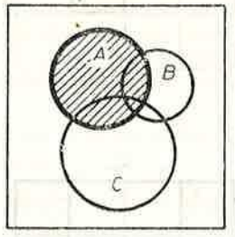


Fig. I.7

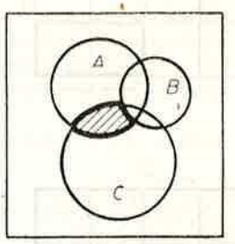


Fig. I.8

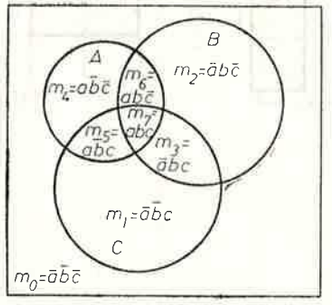


Fig. I.9

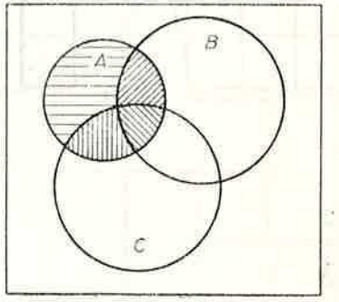


Fig. I.10

se observă că cei patru mintermeni corespund celor patru zone m_4, m_5, m_6 și m_7. Reuniunea acestor patru zone (fig. I.10) formează mulțimea corespunzătoare variabilei a, deci

$$a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c} + abc = a.$$

Utilizarea diagramelor Euler-Venn devine dificilă dacă expresiile conțin mai mult de trei variabile.

Diagramele Veitch-Karnaugh reprezintă rearanjarea diagramelor Euler-Venn sub formă de tablou. În cazul diagramelor Veitch-Karnaugh mulțimile corespunzătoare variabilelor booleene sînt reprezentate de dreptunghiuri. În figura I.11 este prezentată diagrama Veitch-Karnaugh pentru patru variabile. Se observă că fiecare căsuță a diagramei corespunde unui mintermen. În figura I.12 sînt prezentate diagramele Veitch-Karnaugh corespunzătoare termenilor a, $\bar{d}$, $b\bar{c}$, $\bar{b}c$, bcd, $a\bar{b}\bar{d}$. Se observă că mulțimile corespunzătoare unor termeni sînt reprezentate prin unul sau mai multe pătrate.

O proprietate importantă a diagramelor Veitch-Karnaugh este că mintermenii care nu diferă decît printr-un factor sînt reprezentați pe diagramă

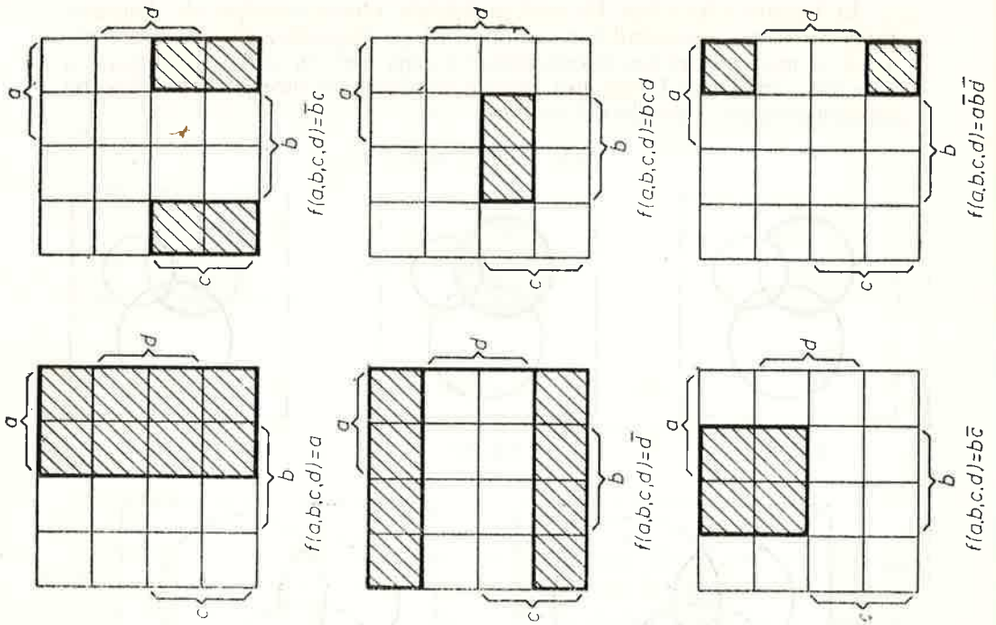


Fig. I.12

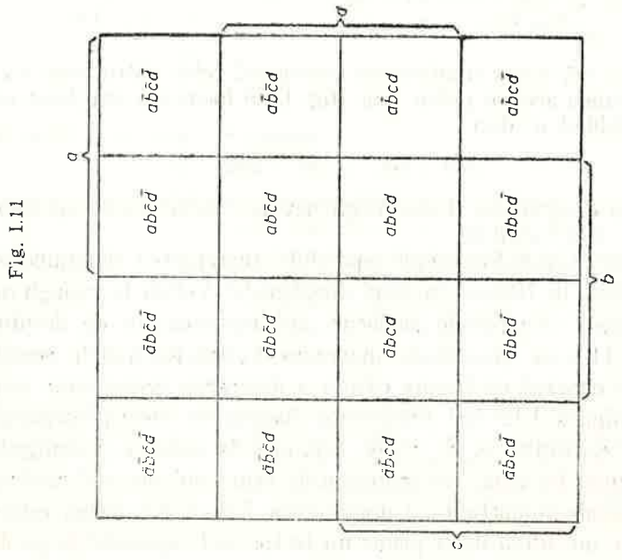


Fig. I.11

prin pătrate alăturate. Pentru aceasta trebuie însă să se considere că laturile opuse ale diagramei sînt confundate (sau suprapuse). Reciproca acestei proprietăți este de asemenea adevărată. Există însă o proprietate mai generală, deoarece și termenii care nu diferă decît prin complementul unei variabile (abc și $ab\bar{c}$, $\bar{b}d$ și $\bar{b}\bar{d}$) se reprezintă pe diagramă prin dreptunghiuri adiacente. Orice dreptunghi, în afara celor cu latura de trei unități corespunde unui termen. Reciproca acestei proprietăți nu reprezintă altceva decît faptul că doi termeni care corespund la două dreptunghiuri alăturate egale pot fi reduși la un singur termen, care corespunde dreptunghiului format prin alăturarea celor două dreptunghiuri. Această proprietate stă la baza metodei de simplificare a funcțiilor booleene cu ajutorul diagramelor Veitch-Karnaugh.

Să considerăm ca exemplu funcția definită prin tabelul I.12, a cărei formă canonică disjunctivă este

$$f(a,\ b,\ c,\ d) = \bar{a}b\bar{c}d + \bar{a}bcd + a\bar{b}c\bar{d} + a\bar{b}cd + ab\bar{c}d + abc\bar{d} + abcd.$$

a	b	c	d	$f(a,\ b,\ c,\ d)$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Tabelul I.12

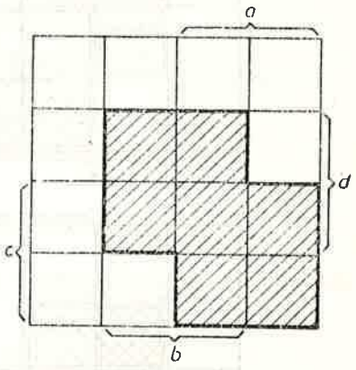
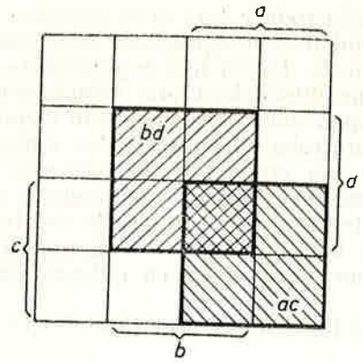


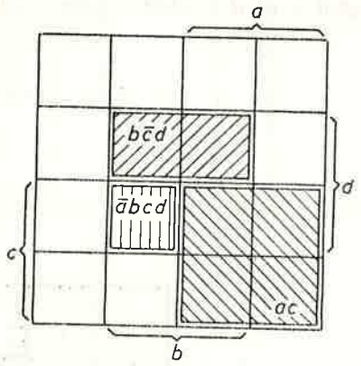
Fig. I.13

În figura I.13 este prezentată diagrama Veitch-Karnaugh corespunzătoare acestei funcții.

Simplificarea funcției corespunde cu acoperirea mulțimii corespunzătoare expresiei respective cu un număr minim de dreptunghiuri cît mai mari. În figura I.14 sînt prezentate trei posibilități de acoperire, corespunzătoare expresiilor $ac + bd$, $ac + b\bar{c}d + \bar{a}bcd$, $b\bar{c}d + \bar{a}bd + ab\bar{c} + ac\bar{d} + abcd$. Se observă că prima expresie, care corespunde acoperirii cu două pătrate mai mari, este cea mai simplă.

Diagrama Veitch-Karnaugh poate fi considerată și ca un tabel de valori cu două dimensiuni al unei funcții. Pentru aceasta, diagrama se rescrie ca în figura I.15. Valorile funcției se vor trece în căsuțele diagramei, considerînd valorile variabilelor funcției drept „coordonate" pentru stabilirea poziției în tabel. Funcția din exemplul anterior va avea diagrama prezentată în figura I.16. Se observă că această diagramă este asemănătoare cu diagrama





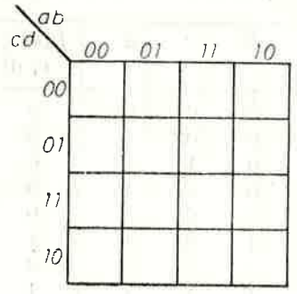


Fig. I.15

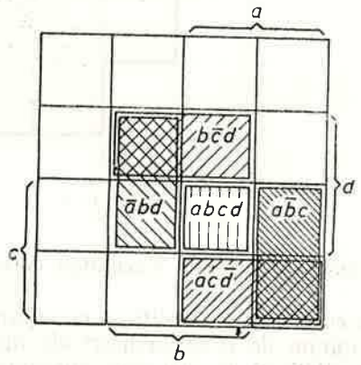


Fig. I.14

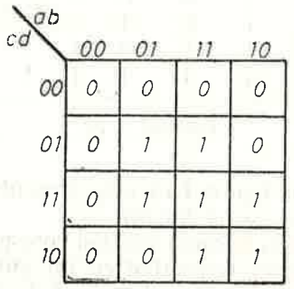


Fig. I.16

clasică. Avantajul acestei forme a diagramelor Veitch-Karnaugh este că deşi reprezintă un tabel de valori, poate servi în acelaşi timp la simplificarea funcţiei.

Diagramele Veitch-Karnaugh se utilizează de obicei pentru funcţiile booleene cu maximum cinci variabile (foarte rar pentru şase variabile).

Diagramele Veitch-Karnaugh pentru o funcţie cu cinci variabile arată ca în figura I.17.

Pentru a putea efectua simplificări pe această diagramă se consideră că pătratele simetrice faţă de axa de simetrie a diagramei sînt adiacente. În figura I.17 funcţia reprezentată este:

$$f(a, b, c, d, e) = bc + a\bar{c}d + \bar{a}\bar{b}\bar{c}(\bar{d} + e).$$

ab \ cd	00	01	11	10	10	11	01	00
00	1	0	0	0	0	0	0	1
01	0	0	1	1	1	1	0	1
11	0	1	1	0	0	1	1	0
10	0	1	1	0	0	1	1	0
	e = 0				e = 1			

Fig. I.17

I.4.3. **Exerciţii**

1. Să se simplifice prin calcul boolean expresiile

   a) $(a\bar{b} + b)(\overline{a + b})$; b) $x_2(\overline{x_3 + x_4}) + x_1x_2(\overline{x_3 + x_4})$;

   c) $x + \bar{z} + y(\overline{x + \bar{z}}) + (x + \bar{z})(x + y + \bar{z})$; d) $\bar{a} + \bar{b} + ab\bar{c}$.

2. Să se simplifice prin calcul boolean complementara expresiilor:

   a) $a + b + \bar{c}\bar{d}$; b) $\overline{x\bar{y}} + z$; c) $\overline{x_1x_2 + \bar{x}_3 + \bar{x}_4}$;

   d) $a + b(\bar{c} + \bar{d})$; e) $x_1(x_2x_3 + \bar{x}_4) + x_1(\bar{x}_2\bar{x}_3 + x_4)$.

3. Să se găsească prin calcul boolean o formă mai simplă pentru expresiile următoarelor funcţii:

   a) $f(x, y, z) = \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + \bar{x}yz$;
   b) $g(a, b, c) = (\bar{a} + b + \bar{c})(a + \bar{b} + c)(\bar{a} + \bar{b} + \bar{c})$;
   c) $h(x_1, x_2, x_3) = x_1x_2 + \bar{x}_1x_3 + x_2x_3$.

4. Să se găsească prin calcul boolean o formă normală disjunctivă mai simplă pentru funcţiile:

   a) $f(x, y, z) = xyz + xy\bar{z} + \bar{x}\bar{y} + \bar{y}$;
   b) $g(x_1, x_2, x_3) = (x_1x_2 + x_1\bar{x}_3)(\bar{x}_2x_3 + x_2x_3)$;
   c) $f(a, b, c) = (a + b)(\overline{\bar{a} + b})(\bar{a} + \bar{b})$.

5. Folosind artificii de calcul boolean, să se simplifice expresiile:

   a) $ac + b\bar{c} + ab$;
   b) $\bar{a}b\bar{c}\bar{d} + ab\bar{c}\bar{d} + \bar{a}bcd + ab\bar{c}d$;
   c) $xy + x\bar{y}z + y(\bar{x} + z) + \bar{y}\bar{z}$.

6. Funcţia majoritară $M(x, y, z)$ se defineşte ca fiind funcţia booleană care are valoarea 1 dacă cel puţin două dintre argumentele sale sînt egale cu 1.
   a) Să se alcătuiască tabelul de valori al acestei funcţii.
   b) Să se afle forma canonică disjunctivă şi forma canonică conjunctivă a acestei funcţii.
   c) Să i se determine prin calcul boolean cea mai simplă formă normală disjunctivă.
   d) Să se arate că $M[a, b, M(c, d, e)] = M[M(a, b, c), d, M(a, b, e)]$.

7. Pentru funcțiile din următorul tabel să se afle:
a) forma canonică disjunctivă;
b) forma canonică conjunctivă;
c) cea mai simplă formă normală disjunctivă.

x_1	x_2	x_3	f_1	f_2	f_3
0	0	0	1	1	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	1	1	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	1	1	0

Tabelul I.13

8. Să se indice pentru fiecare din funcțiile ale căror diagrame Euler-Venn sînt reprezentate în figura I.18 forma canonică disjunctivă și să se simplifice.

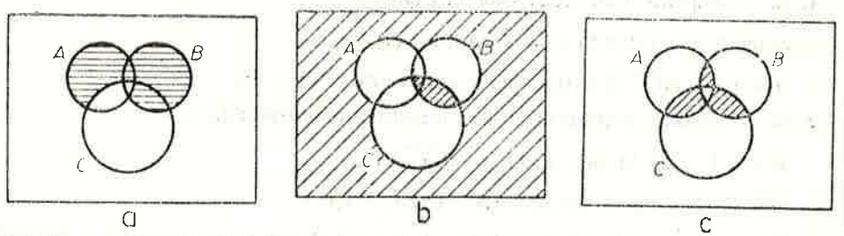


Fig. I.18

9. Să se alcătuiască diagramele Euler-Venn pentru următoarele funcții de trei variabile date prin dezvoltările lor și să se simplifice:

a) $m_1 \cup m_3 \cup m_5 \cup m_7$; b) $m_2 \cup m_3 \cup m_4 \cup m_5$;

c) $m_0 \cup m_1$; d) $m_1 \cup m_2 \cup m_3 \cup m_4 \cup m_5 \cup m_6$,

unde $m_0, m_1, \ldots, m_7$ corespund notațiilor din figura I.9.

10. Să se construiască diagramele Veitch-Karnaugh pentru următoarele funcții booleene:
a) $f(a, b, c) = a + b\bar{c} + ac + abc$;

b) $f(x_1, x_2, x_3, x_4) = x_1\bar{x}_2x_3 + x_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3$;

c) $f(x, y, z) = x + y + xy + yz + xz + xyz$;

d) $g(x, y, z, w) = x\bar{y} + x\bar{z} + yz + \bar{x}\bar{y}$;

e) $f(a, b, c, d) = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}\bar{d} + \bar{a}bcd + a\bar{b}\bar{c}\bar{d} + a\bar{b}cd + abcd$;

f) $g(x_1, x_2, x_3, x_4) = (x_1 + \bar{x}_2)x_3 + (x_2 + \bar{x}_4)x_1$.

11. Aflați funcțiile reprezentate prin diagramele Veitch-Karnaugh din figura I.19.

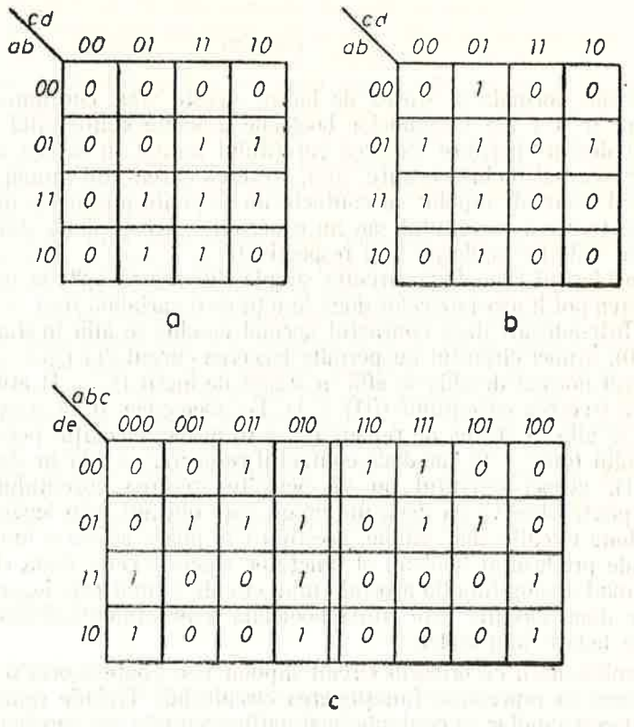


Fig. I.19

12. Folosind diagrama Veitch-Karnaugh, să se simplifice următoarele funcții:
a) $f(a, b, c, d) = a\bar{b} + \bar{a}\bar{c}d + c + b\bar{c}d$;
b) $f(x, y, z) = (x + \bar{y} + \bar{z})(\bar{x} + y)(\bar{y} + z)$;
c) $g(x, y, z, w) = \bar{x}yz + xyw + xy\bar{z}w + x\bar{z}\bar{w} + \bar{y}\bar{z}w + \bar{x}\bar{y}zw$;
d) $h(x_1, x_2, x_3, x_4) = x_2\bar{x}_3\bar{x}_4 + x_1x_2\bar{x}_4 + \bar{x}_1x_2x_3\bar{x}_4$;
e) $f(a, b, c, d) = a\bar{c}d + ab\bar{c}d + a\bar{b}\bar{c} + \bar{a}bd + bcd$.

I.5. Realizarea fizică a funcțiilor booleene

I.5.1. **Circuite cu contacte**

O primă aplicație a teoriei funcțiilor booleene este reprezentată de studiul circuitelor dipolare cu contacte. Aceste circuite sînt formate prin legarea în serie sau în paralel a unor contacte care sînt de două tipuri: contacte normal deschise și contacte normal închise.

Fiecare contact poate fi pus în corespondență cu o variabilă booleană. Contactele normal închise sînt puse în corespondență cu complementele variabilelor booleene. Un contact se poate afla în două stări: starea de

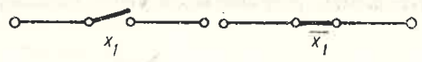


Fig. I.20

repaus sau normală și starea de lucru. Aceste stări corespund, respectiv, valorilor 0 și 1 ale variabilelor booleene asociate contactului. Un contact normal deschis permite trecerea curentului numai în starea de lucru. Un contact normal închis permite, însă, trecerea curentului numai în starea de repaus. Un circuit dipolar cu contacte nu se poate afla decît în două stări: permite trecerea curentului sau nu o permite. Aceste două stări vor fi asociate cu valorile booleene 1 și respectiv 0.

Considerînd cele două circuite simple din figura I.20 se poate observa că acestea pot fi asociate celor două funcții de o variabilă $f(x_1) = x_1$ și $g(x_1) = \bar{x}_1$. Într-adevăr, dacă contactul normal deschis se află în starea de repaus ($x_1 = 0$), atunci circuitul nu permite trecerea curentului ($f(0) = 0$), iar dacă contactul normal deschis se află în starea de lucru ($x_1 = 1$) atunci circuitul permite trecerea curentului ($f(1) = 1$). De asemenea, dacă contactul normal închis se află în starea de repaus ($x_1 = 0$) atunci circuitul permite trecerea curentului ($g(0) = 1$), iar dacă contactul respectiv se află în starea de lucru ($x_1 = 1$), atunci circuitul nu va permite trecerea curentului ($g(1) = 0$).

Se poate observa că dacă un circuit este obținut prin legarea în serie a altor două circuite mai simple, acestuia i se poate asocia o funcție care corespunde produsului boolean al funcțiilor asociate celor două circuite.

În mod analog funcția asociată unui circuit format prin legarea în paralel a altor două circuite este suma booleană a funcțiilor asociate celor două circuite legate în paralel.

Rezultă, deci, că oricărui circuit dipolar i se poate asocia o funcție booleană care să reprezinte funcționarea circuitului. Trebuie remarcat că, într-un circuit dipolar cu contacte, mai multe contacte pot corespunde aceleiași variabile sau complementului acesteia.

De exemplu, circuitului din figura I.21 îi corespunde funcția $f(x, y) = x + x\bar{y}$, iar celui din figura I.22 i se asociază funcția $f(x_1, x_2, x_3, x_4) = (x_1 + x_2)(x_1x_3 + (\bar{x}_1 + x_2)x_4)$

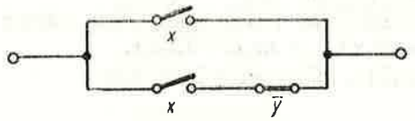


Fig. I.21

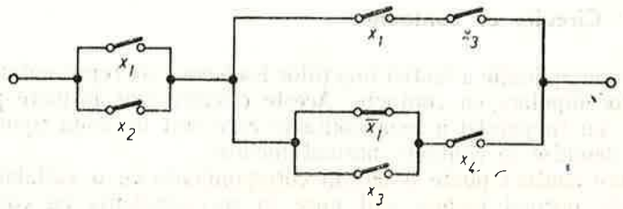


Fig. I.22

Reciproc, orice funcție booleană poate fi realizată printr-un circuit dipolar cu contacte, adică se poate alcătui un circuit dipolar cu contacte a cărui funcționare să fie reprezentată de funcția dată.

Realizarea funcțiilor booleene prin circuite dipolare cu contacte reprezintă o modalitate de utilizare practică a teoriei funcțiilor booleene.

Este cunoscut că funcționarea celor mai multe circuite de automatizare, precum și funcționarea calculatoarelor electronice se bazează pe utilizarea numerelor binare, care pot fi reprezentate cu numai două cifre, 0 și 1. Funcțiile care reprezintă comportarea în funcționare a circuitelor respective sînt funcții booleene. Aceste circuite pot fi deci construite, realizînd fizic funcțiile booleene corespunzătoare.

Simplificarea unei funcții booleene reprezintă în acest caz realizarea aceleiași funcții printr-un număr cît mai mic de contacte.

În practică, circuitele cu contacte sînt folosite de obicei sub formă de circuite cu relee și contacte. Releele sînt necesare în cadrul acestor circuite pentru a acționa contactele. Un releu va acționa toate contactele corespunzătoare unei variabile, astfel că valorile variabilei pot fi asociate cu cele două stări, de conducere a curentului sau nu, corespunzătoare circuitului de comandă a releului. Se dă astfel posibilitatea de a se reprezenta în mod omogen valorile 1 și 0 prin starea de conducere sau nu a curentului prin diferite circuite. Circuitele cu relee și contacte au fost înlocuite treptat prin circuite logice alcătuite din tuburi electronice, apoi prin circuite alcătuite din tranzistoare, iar în ultimul timp prin circuite integrate.

I.5.2. Circuite logice simple

Circuitele logice sînt realizate sub forma unor circuite multipolare (cu mai multe borne). Un circuit logic are una sau mai multe borne de intrare și una sau mai multe borne de ieșire. La bornele de intrare se aplică semnale (în general impulsuri de tensiune electrică), care reprezintă variabile booleene și pot lua două valori (de exemplu, o tensiune de 4 V poate reprezenta valoarea logică „1", iar o tensiune de 0 V valoarea logică „0"). La bornele de ieșire apar semnale, care corespund valorilor unor funcții booleene de variabilele de la intrare. Aceste circuite logice se vor numi combinaționale. Există posibilitatea ca funcțiile booleene să nu depindă numai de valorile actuale ale variabilelor de la intrare ci și de valorile precedente ale acestor variabile. În acest caz circuitele logice se vor numi secvențiale sau „cu memorie".

În manualul de față nu vom studia decît circuitele logice combinaționale.

În cazul general, un circuit combinațional are n borne de intrare, corespunzătoare unor variabile booleene $x_1, x_2, \ldots, x_n$ și m borne de ieșire, corespunzătoare unor funcții $y_1(x_1, \ldots, x_n), \ldots, y_m(x_1, \ldots, x_n)$. Un astfel de circuit se reprezintă ca în figura I.23, fără a se descrie structura sa internă.

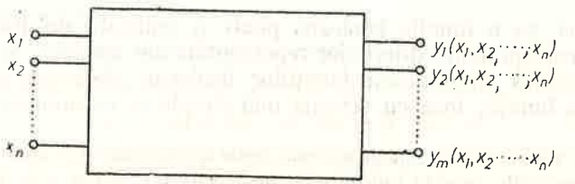


Fig. I.23

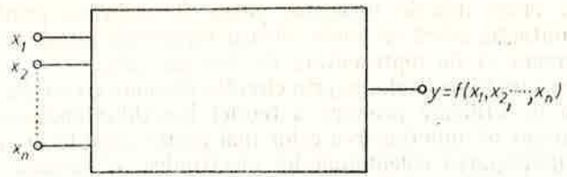


Fig. I.24

O proprietate importantă a circuitelor logice este că acestea pot fi combinate pentru a forma circuite logice mai complexe. Astfel, bornele de ieşire ale unor circuite pot fi cuplate la bornele de intrare ale altor circuite. Această proprietate permite ca un circuit logic mai complex să poată fi realizat din circuite logice mai simple.

Un circuit combinaţional cu mai multe borne de ieşire poate fi conceput ca fiind format din mai multe circuite care au fiecare cîte o singură bornă de ieşire şi care corespund fiecare la cîte o funcţie. Un circuit combinaţional cu o singură bornă de ieşire va fi denumit circuit logic simplu (fig. I.24). Funcţionarea unui circuit logic simplu este descrisă printr-o funcţie booleană de variabilele de la intrare. Putem spune astfel că un circuit logic simplu realizează funcţia booleană corespunzătoare. Dacă funcţia booleană este o funcţie elementară de două variabile, circuitul logic simplu corespunzător care o realizează va fi denumit circuit logic elementar. Există deci trei circuite logice elementare, corespunzătoare celor trei operaţii de bază ale algebrelor booleene, şi anume circuitul ŞI pentru operaţia „$\cdot$“, circuitul SAU pentru operaţia „$+$“ şi circuitul NU pentru operaţia de complementare. Aceste circuite logice elementare au o reprezentare convenţională specială care este arătată în figura I.25.

Deoarece orice funcţie booleană poate fi reprezentată cu ajutorul celor trei funcţii elementare, corespunzătoare celor trei operaţii „$\cdot$“, „$+$“ şi „$^-$“, rezultă că orice circuit logic simplu poate fi realizat prin combinarea unor circuite logice elementare. Reprezentarea unei funcţii booleene cu ajutorul funcţiilor elementare poate fi considerată drept schema de combinare a circuitelor logice elementare.

De exemplu, funcţia

$$f(x_1, x_2, x_3) = x_1x_2 + \bar{x}_2\bar{x}_3$$

care poate fi reprezentată sub forma

$$((x_1 \cdot x_2) + ((\bar{x}_2) \cdot (\bar{x}_3))),$$

este realizată de circuitul logic simplu din fig. I.26.

Dacă reprezentăm însă aceeaşi funcţie prin

$$f(x_1, x_2, x_3) = x_1x_2 + \overline{x_2 + x_3},$$

realizarea funcţiei va fi diferită, corespunzînd circuitului logic din figura I.27.

Rezultă deci, că o funcţie booleană poate fi realizată de diferite circuite logice simple, corespunzînd diferitelor reprezentări ale acesteia. Se poate acum înţelege importanţa simplificării funcţiilor booleene, deoarece astfel se pot realiza aceleaşi funcţii, însă cu scheme mai simple şi economice.

În I.2.3. s-a arătat că orice funcţie booleană poate fi reprezentată cu ajutorul unei singure funcţii de două variabile, această funcţie fiind fie funcţia lui Sheffer $f_{14}(x, y) = \bar{x} + \bar{y}$, fie funcţia lui Peirce $f_8(x, y) = \bar{x}\bar{y}$. Există circuite logice simple care realizează aceste funcţii.

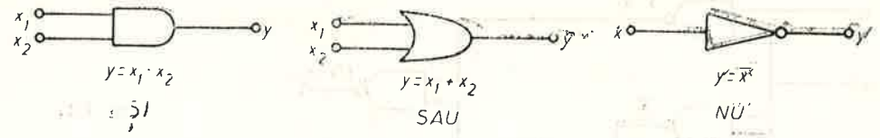


Fig. I.25

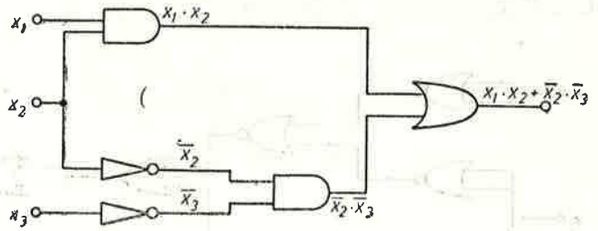


Fig. I.26

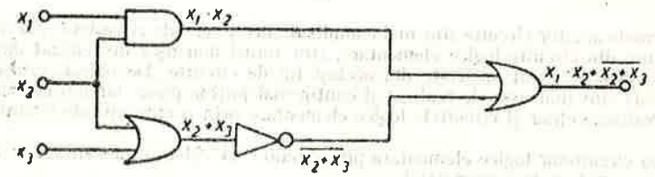


Fig. I.27

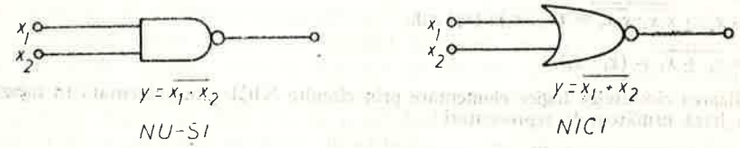


Fig. I.28

Aceste circuite vor fi denumite circuite logice universale. Rezultă, deci, că orice circuit logic simplu poate fi realizat printr-o combinaţie de circuite logice universale. Circuitele logice universale sînt simbolizate ca în figura I.28. Aceste circuite poartă denumiri proprii, particulare. Astfel circuitul care realizează funcţia lui Sheffer se numeşte circuit NU-ŞI, iar cel care realizează funcţia lui Peirce se numeşte circuitul NICI.

Funcţia considerată anterior poate fi reprezentată astfel:

— prin funcţia lui Sheffer (notată cu „$|$“)

$$f(x_1, x_2, x_3) = (x_1 \mid x_2) \mid ((x_2 \mid x_2) \mid (x_3 \mid x_3))$$

— prin funcţia lui Peirce (notată cu „$\downarrow$“)

$$g(x_1, x_2, x_3) = ((x_1 \downarrow x_1) \downarrow (x_2 \downarrow x_2)) \downarrow (x_3 \downarrow x_3),$$

iar $f(x_1, x_2, x_3) = g(x_1, x_2, x_3) \downarrow g(x_1, x_2, x_3)$.

Circuitele logice simple corespunzătoare acestor reprezentări pot fi văzute în figura I.29 şi figura I.30.

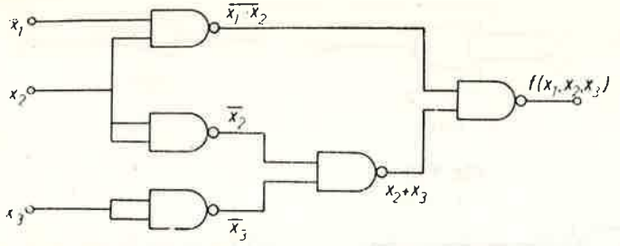


Fig. I.29

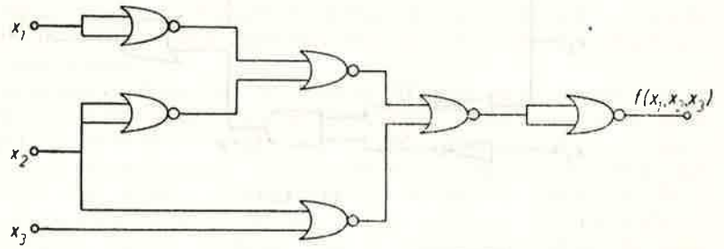


Fig. I.30

Deşi schemele acestor circuite sînt mai complicate decît cele ale circuitelor din figurile I.26 şi I.27, compuse din circuite logice elementare, sînt totuşi mai uşor de realizat din punct de vedere tehnic, deoarece sînt alcătuite din acelaşi tip de circuite. De obicei, însăşi circuitele logice universale sînt mai uşor de realizat şi conţin mai puţine piese, iar uneori este mai economic să se realizeze chiar şi circuitele logice elementare prin combinaţii ale circuitelor logice universale.

Realizarea circuitelor logice elementare prin circuite NU-ŞI este prezentată în figura I.31 şi are la bază următoarele reprezentări :

$$x_1 \cdot x_2 = \overline{\overline{x_1 x_2} \cdot \overline{x_1 x_2}} = (x_1 \mid x_2) \mid (x_1 \mid x_2),$$

$$x_1 + x_2 = \overline{\overline{x_1 x_1} \cdot \overline{x_2 x_2}} = (x_1 \mid x_1) \mid (x_2 \mid x_2),$$

$$\overline{x_1} = \overline{x_1 + x_1} = (x_1 \mid x_1).$$

Realizarea circuitelor logice elementare prin circuite NICI este prezentată în figura I.32 şi are la bază următoarele reprezentări :

$$x_1 \cdot x_2 = \overline{\overline{x_1 + x_1} + \overline{x_2 + x_2}} = (x_1 \downarrow x_1) \downarrow (x_2 \downarrow x_2),$$

$$x_1 + x_2 = \overline{\overline{x_1 + x_2} + \overline{x_1 + x_2}} = (x_1 \downarrow x_2) \downarrow (x_1 \downarrow x_2),$$

$$\overline{x_1} = \overline{x_1 + x_1} = x_1 \downarrow x_1.$$

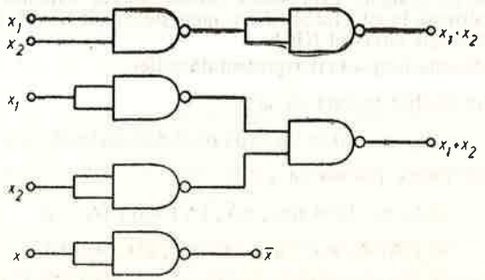


Fig. I.31

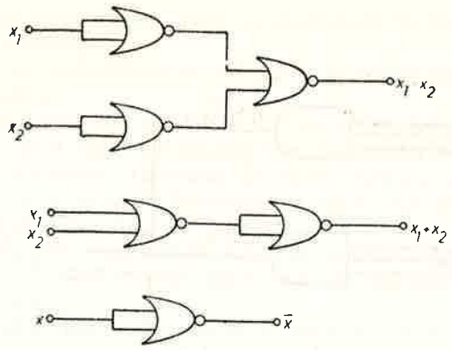


Fig. I.32

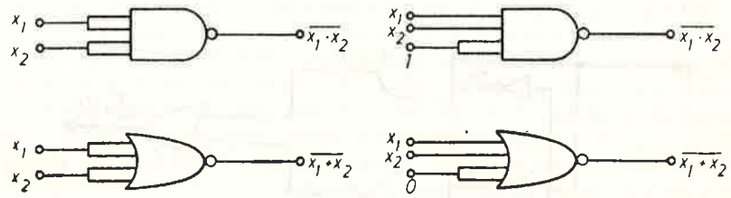


Fig. I.33

În tehnică sînt de asemenea uşor de realizat circuite ŞI, SAU, NICI sau NU-ŞI cu mai mult decît două borne de intrare. Aceste circuite au aceleaşi simboluri ca şi circuitele corespunzătoare cu numai două borne de intrare. Un astfel de circuit poate fi realizat oricînd în locul unui circuit similar cu mai puţine borne de intrare, dacă la bornele de intrare suplimentare se aplică fie semnale avînd o valoare logică constantă („1" pentru circuitele „ŞI" şi „NU-ŞI" şi „0" pentru „SAU" şi „NICI"), fie unul dintre semnalele de la celelalte borne. În figura I.33 se prezintă exemple de utilizare a unui circuit cu patru borne de intrare pentru realizarea unei funcţii booleene cu două variabile.

Circuitele logice simple cu un număr mai mare de borne de intrare de tip ŞI şi SAU pot fi utilizate la realizarea funcţiilor booleene reprezentate sub o formă normală.

De exemplu, funcţia $f(x_1, x_2, x_3) = x_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 x_3$ poate fi realizată cu circuitul din figura I.34.

În practică pot apare situaţii care să necesite aflarea funcţiei booleene realizate de un circuit logic simplu a cărui schemă (structură) este cunoscută.

Pentru a obţine expresia funcţiei booleene realizate de un circuit se procedează astfel : se identifică variabilele booleene corespunzătoare bornelor de intrare ale circuitului ; se alcătuiesc expresiile booleene corespunzătoare acelor circuite elementare pentru care s-au identificat variabilele de la intrare ; aceste expresii constituie intrările altor circuite elementare ; se obţin astfel expresii din ce în ce mai complexe, iar în final expresia căutată, corespunzătoare ieşirii circuitului.

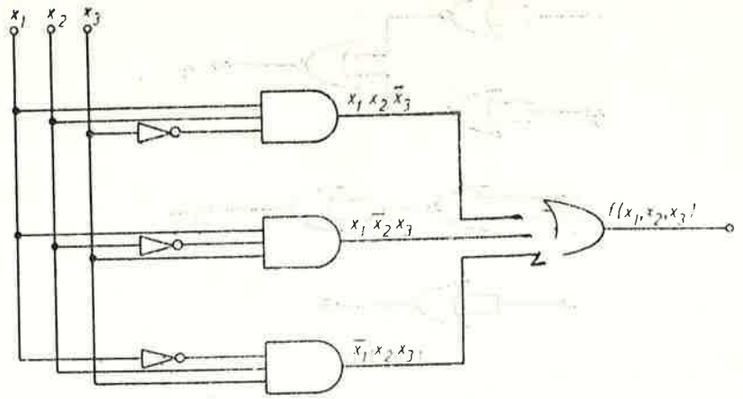


Fig. I.34

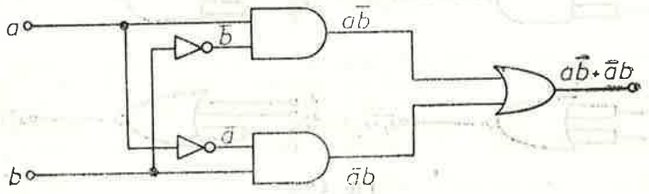


Fig. I.35

De exemplu, circuitul din figura I.35 are ca variabile de intrare a și b. Dacă în dreptul ieșirii fiecărui circuit notăm expresiile booleene corespunzătoare, vom scrie pe rînd, $\bar{a}$, $\bar{b}$, $a\bar{b}$, $\bar{a}b$, $a\bar{b} + \bar{a}b$.

1.5.3. Proiectarea circuitelor logice

Circuitele logice reprezintă o parte importantă din circuitele unui calculator electronic. Ele realizează diferite expresii booleene care sînt necesare pentru funcționarea calculatorului. Este cunoscut că funcționarea calculatorului se bazează pe transformarea unor semnale binare, ale căror valori „0" sau „1" au fiecare o anumită semnificație. De exemplu, un număr se reprezintă întotdeauna în baza 2, astfel că cifrele sale nu sînt decît 0 sau 1. Un număr cu cinci cifre binare se reprezintă cu ajutorul a cinci semnale binare, cîte un semnal pentru fiecare cifră. Semnul unui număr poate fi de asemenea reprezentat printr-un semnal binar, considerînd că valoarea 0 corespunde semnului +, iar valoarea 1 semnului —.

Transformarea semnalelor binare este efectuată conform unor reguli bine stabilite, în funcție de rezultatul urmărit. Adunarea a două numere binare se reduce la o transformare a semnalelor corespunzătoare celor doi operanzi, pentru a se obține semnalele corespunzătoare sumei, conform regulilor de adunare a două numere.

O activitate importantă din cadrul etapei de proiectare a unui calculator electronic o constituie proiectarea circuitelor logice din care este alcătuit.

Proiectarea unui circuit logic se desfășoară în mai multe faze, descrise pe scurt în continuare:

— stabilirea variabilelor de intrare și de ieșire precum și a semnificației valorilor acestora;

— determinarea funcției booleene (sau a funcțiilor, în cazul mai multor variabile de ieșire), care corespunde transformărilor necesare ale variabilelor de intrare și alcătuirea tabelului de valori corespunzător;

— aflarea unei expresii booleene, de obicei sub forma canonică disjunctivă, corespunzătoare fiecărei funcții;

— aflarea celei mai simple expresii pentru fiecare funcție booleană;

— alcătuirea unui circuit logic care să realizeze funcțiile booleene în forma simplificată.

Este posibil ca unele etape să nu fie necesare în anumite cazuri. De exemplu, dacă tabelul de valori al funcției este alcătuit sub formă de diagramă Veitch-Karnaugh, nu este necesară aflarea unei expresii oarecare a funcției, ci se poate trece direct la aflarea celei mai simple expresii a funcției respective.

Proiectarea unui circuit cu contacte se face urmînd aceleași faze astfel ca în ultima fază să se alcătuiască schema circuitului respectiv.

Pentru exemplificare se va prezenta proiectarea unui circuit comparator.

Circuitul comparator care trebuie proiectat este destinat să compare două numere de cîte două cifre, indicînd pe cel mai mare dintre acestea.

Variabilele de intrare sînt în număr de patru. Alegem, de exemplu, ca x_1 și x_2 să reprezinte cifrele binare ale primului număr, iar x_3 și x_4 pe cele ale numărului al doilea. Semnalul de ieșire va indica numărul cel mai mare astfel, dacă primul număr este cel mai mare, semnalul de ieșire va avea valoarea „1", iar dacă al doilea număr este cel mai mare, semnalul de ieșire va avea valoarea „0". Dacă cele două numere sînt egale, convenim să considerăm al doilea număr ca fiind cel mai mare. În tabelul I.14 se indică valorile funcției circuitului comparator considerînd că numerele sînt reprezentate în baza 10, iar în tabelul I.15 funcția circuitului este definită ca o funcție booleană. Diagrama Veitch-Karnaugh a acestei funcții este prezentată în figura I.36.

n_1	n_2	f
0	0	0
0	1	0
0	2	0
0	3	0
1	0	1
1	1	0
1	2	0
1	3	0
2	0	1
2	1	1
2	2	0
2	3	0
3	0	1
3	1	1
3	2	1
3	3	0

Tabelul I.14

x_1	x_2	x_3	x_4	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Tabelul I.15

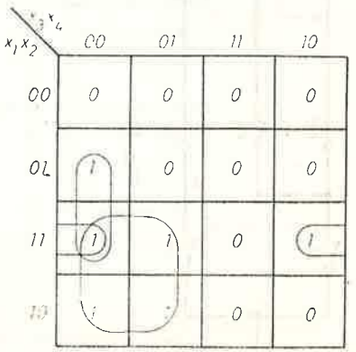


Fig. I.36

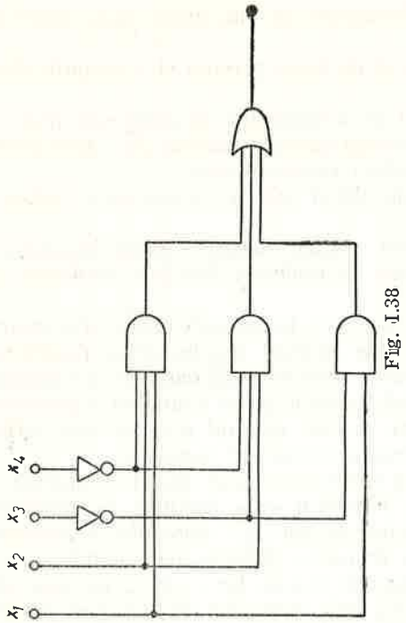


Fig. I.38

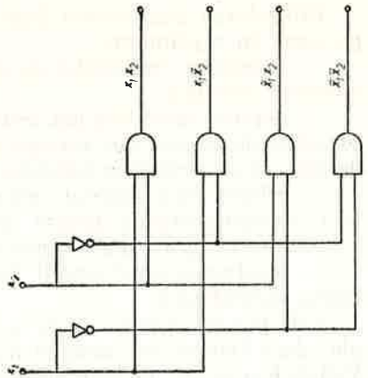


Fig. I.41

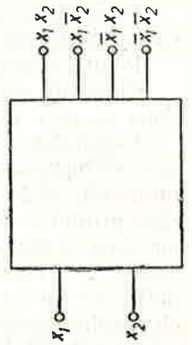


Fig. I.40

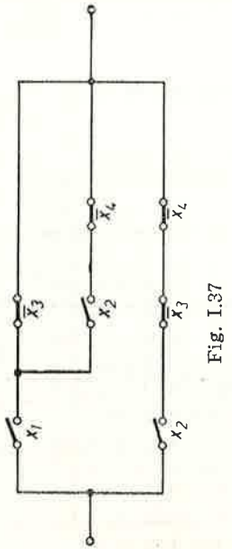


Fig. I.37

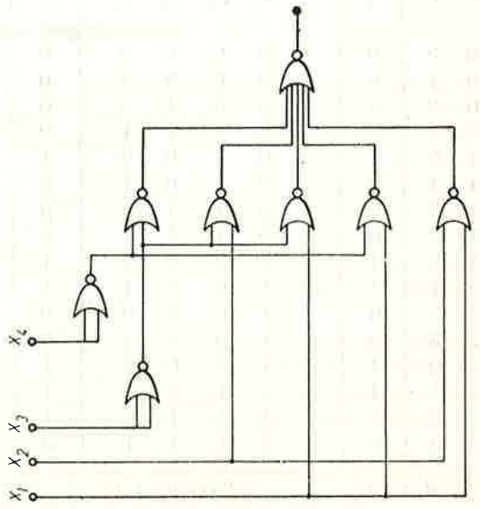


Fig. I.39

Forma canonică disjunctivă a acestei funcții este:

$$f(x_1,x_2,x_3,x_4) = \bar{x}_1x_2\bar{x}_3\bar{x}_4+x_1\bar{x}_2\bar{x}_3\bar{x}_4+x_1\bar{x}_2\bar{x}_3x_4+x_1x_2\bar{x}_3\bar{x}_4+x_1x_2\bar{x}_3x_4+x_1x_2x_3\bar{x}_4.$$

Cea mai simplă formă normală disjunctivă este:

$$f(x_1,\ x_2,\ x_3,\ x_4) = x_1\bar{x}_3 + x_1x_2\bar{x}_4 + x_2\bar{x}_3\bar{x}_4,$$

după cum se poate vedea și pe diagrama din figura I.36. Corespunzător acestei forme normale se obține circuitul cu contacte din figura I.37 și circuitul logic din figura I.38.

Dacă se dorește realizarea circuitului logic numai prin circuite NICI, transformînd funcția definită anterior, se obține expresia

$$f(x_1,\ x_2,\ x_3,\ x_4) = \overline{\bar{x}_1\bar{x}_2 + \bar{x}_1x_3 + \bar{x}_1x_4 + \bar{x}_2x_3 + x_3x_4}$$

și circuitul asociat din figura I.39.

I.5.4. **Circuite logice complexe**

În echipamentele electronice sînt utilizate adeseori circuite logice complexe, care realizează simultan mai multe funcții booleene. Aceste funcții, de obicei nu sînt independente, ci se completează reciproc.

Vom analiza în continuare cîteva circuite logice complexe, mai frecvent utilizate.

Decodificatoare

Circuitele decodificatoare sînt destinate să recunoască diferitele combinații de valori ale variabilelor de la intrare. Schematic, un circuit decodificator se reprezintă ca în figura I.40, indicîndu-se pentru fiecare bornă de ieșire combinația de valori ale variabilelor de intrare corespunzătoare. Rezultă, deci, că funcțiile booleene realizate de un astfel de circuit sînt compuse fiecare dintr-un singur mintermen. Se poate astfel observa că pentru orice combinație de valori ale variabilelor de intrare, una și numai una dintre bornele de ieșire va indica valoarea logică „1" (și anume borna de ieșire a funcției compuse din mintermenul asociat respectivei combinații de valori ale variabilelor). În figura I.41 se prezintă un circuit decodificator pentru două variabile.

În practică pot fi necesare uneori circuite decodificatoare care să indice absența unor combinații de valori ale variabilelor de la intrare. Desigur că în acest caz pentru fiecare combinație de valori ale variabilelor de la intrare numai o singură bornă de ieșire va indica valoarea logică „0" (și anume borna de ieșire a funcției, corespunzătoare acelei combinații, deoarece această combinație nu este absentă). Un exemplu de astfel de circuit decodificator cu două variabile este prezentat în figura I.42.

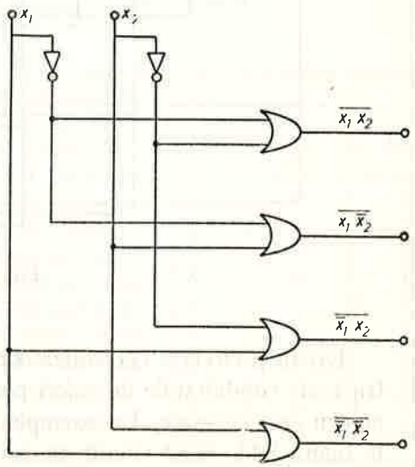


Fig. I.42

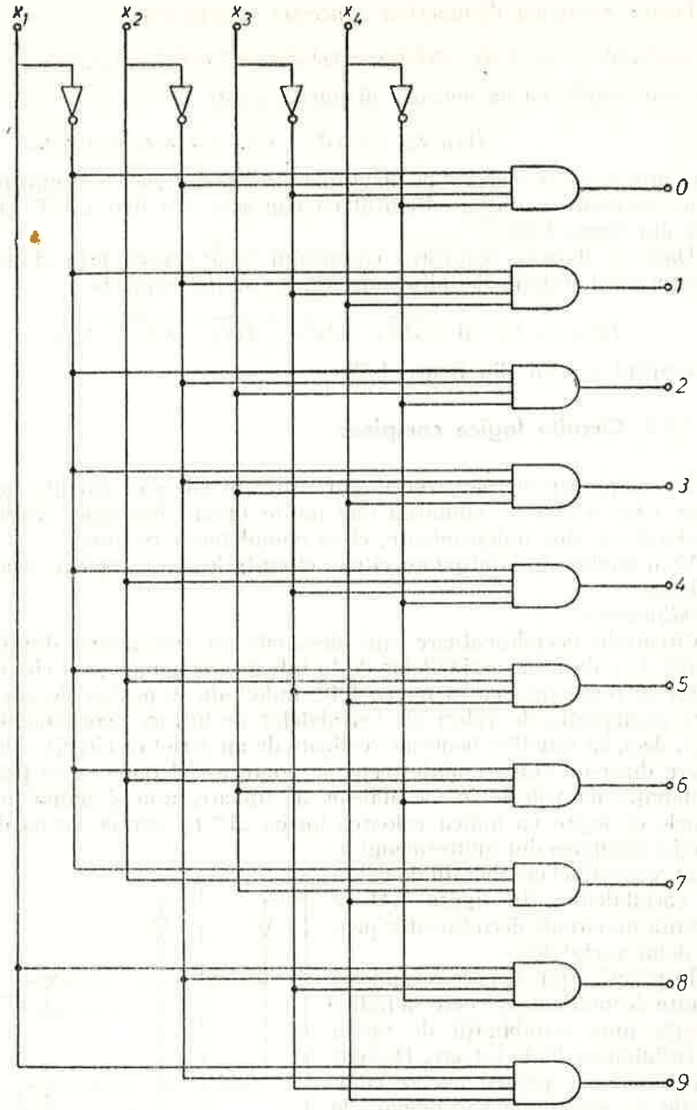


Fig. I.43

Există și circuite decodificatoare reduse, care nu au borne de ieșire pentru toate combinațiile de valori posibile ale variabilelor de intrare, ci numai pentru cele necesare. Un exemplu de astfel de decodificator este prezentat în figura I.43. Acest circuit servește la decodificarea cifrelor zecimale codificate în binar, conform tabelului I.16.

x_1	x_2	x_3	x_4	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

Tabelul I.16

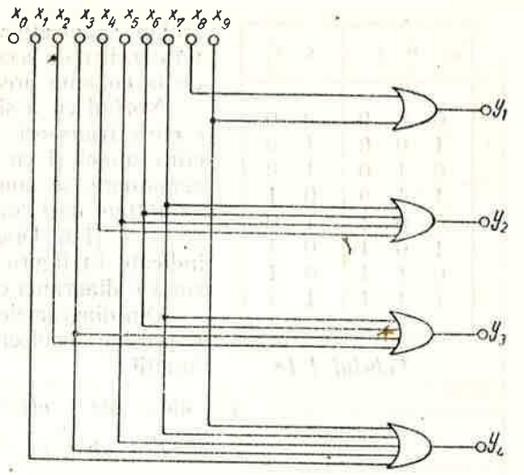


Fig. I.44

Codificatoare

Circuitele codificatoare realizează o operație inversă în raport cu circuitele decodificatoare. Astfel, variabilele de la bornele de intrare ale acestor circuite nu capătă toate combinațiile de valori posibile, deoarece atunci cînd o variabilă are valoarea „1", celelalte variabile nu pot avea decît valoarea „0". Rezultă, deci, că numărul de combinații ale variabilelor de intrare este egal cu numărul variabilelor de intrare. Pentru fiecare dintre aceste combinații se obține la bornele de ieșire o combinație de valori care reprezintă codul semnalului de la intrare. Un circuit codificator este compus din mai multe circuite SAU, cîte unul pentru fiecare bornă de ieșire. Fiecare circuit SAU are mai multe intrări. În figura I.44 este prezentat un circuit codificator care codifică în binar cifrele 0, 1, ..., 9. Codul considerat pentru fiecare cifră este indicat în tabelul I.17. Se observă că borna de intrare corespunzătoare cifrei 0 nu este legată cu alt circuit, deoarece codul necesar este format din valorile 0 pentru toate bornele de ieșire.

Sumatoare

Unul dintre circuitele cele mai importante ale unui calculator electronic îl constituie sumatorul. Acest circuit participă la efectuarea operațiilor aritmetice. Este cunoscut că numerele cu care lucrează un calculator electronic sînt de obicei reprezentate în baza de numerație 2.

Adunarea a două numere binare (în baza 2) se efectuează conform regulilor obișnuite, cifră cu cifră. Un circuit sumator este realizat pe baza circuitului sumator elementar. Circuitul sumator elementar efectuează adunarea a două cifre binare. Rezultatul adunării este alcătuit din două cifre binare: o cifră a sumei și o cifră transport care trebuie adunată la cifrele coloanei alăturate din stînga. Aceasta înseamnă că în realitate un circuit sumator elementar va

	y_1	y_2	y_3	y_4
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Tabelul I.17

a	b	t	s	t'
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

Tabelul I.18

trebui să efectueze adunarea a trei cifre binare (cîte o cifră de la cele două numere și cifra transport de la coloana precedentă).

Notînd cu a și b cifrele celor doi operanzi, cu t cifra transport de la coloana anterioară, cu s cifra sumei și cu t' cifra transport către coloana următoare, se poate alcătui tabelul I.18.

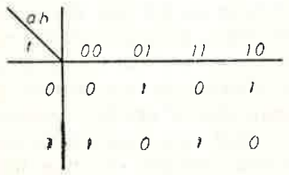
Putem deci considera s și t' ca funcții booleene de a, b și t. Diagramele celor două funcții sînt indicate în figura I.45, iar în figura I.46 se prezintă o diagramă comună pentru cele două funcții.

Din diagramele Veitch-Karnaugh se pot deduce expresiile booleene simplificate ale celor două funcții :

$$s = \bar{a}\bar{b}t + \bar{a}b\bar{t} + a\bar{b}\bar{t} + abt \text{ și}$$

$$t' = ab + at + bt.$$

Folosind aceste expresii, un sumator elementar are schema prezentată în figura I.47.



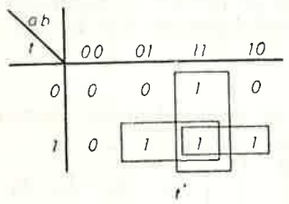


Fig. I.45

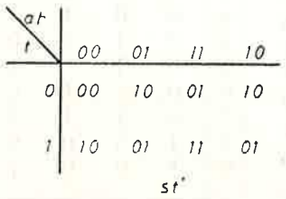


Fig. I.46

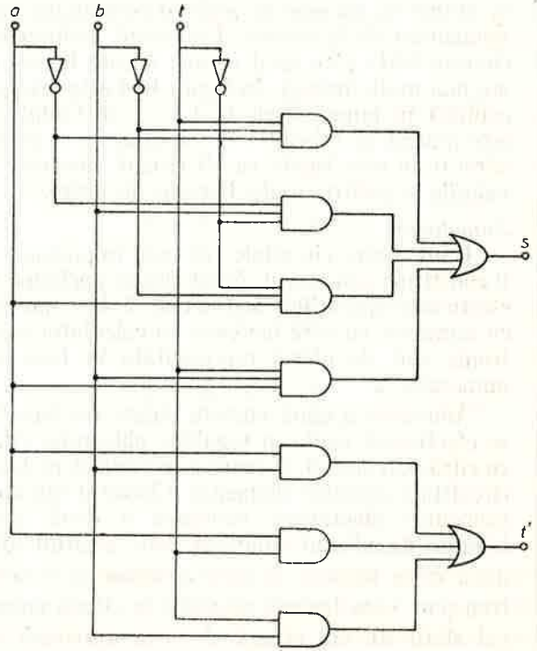


Fig. I.47

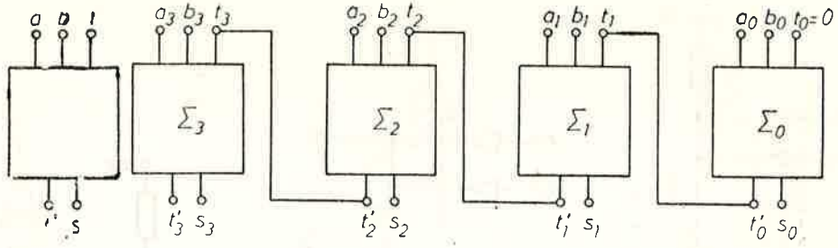


Fig. I.48

Fig. I.49

Simbolic, un sumator elementar poate fi reprezentat ca orice circuit logic (fig. I.48).

Un sumator pentru numere binare cu n cifre este alcătuit din n sumatoare elementare, care adună fiecare cîte o cifră. Circuitele sumatoare elementare sînt interconectate pentru a-și transmite reciproc cifrele de transport respective. Schema unui sumator pentru patru cifre binare este prezentată în figura I.49. Un astfel de sumator este denumit sumator paralel, deoarece cifrele celor două numere sînt introduse simultan.

Dacă este memorată cifra transport, atunci un singur sumator elementar poate fi utilizat (în mod repetat) pentru calculul tuturor cifrelor rezultatului. Un astfel de circuit, denumit sumator serie, nu mai constituie însă un circuit combinațional, ci un circuit secvențial, iar realizarea fizică a sa este mai complicată decît a sumatorului paralel.

I.5.5. **Realizarea practică a circuitelor logice**

Circuitele logice elementare sînt realizate practic utilizînd elemente de circuit obișnuite : rezistențe, condensatoare, diode, tranzistori. Există diferite scheme de circuite electrice care realizează funcții booleene.

Vom prezenta cîteva exemple de scheme simple care realizează operațiile booleene elementare. În cazul acestor scheme se presupune că valoarea logică 0 este reprezentată de o tensiune de 0 V, iar valoarea logică 1 de o tensiune de +4 V.

În figura I.50 este prezentată schema unui circuit ȘI. Analizînd funcționarea circuitului, se observă că dacă una dintre bornele de intrare are tensiunea 0 V, dioda corespunzătoare se află în stare de conducție, astfel că borna de ieșire are de asemenea tensiunea 0 V. Deci, borna de ieșire va avea tensiunea de +4 V numai dacă toate bornele de intrare vor avea de asemenea o tensiune de +4 V, toate diodele fiind astfel blocate.

În figura I.51 este indicată schema unui circuit SAU. În acest caz, oricare dintre bornele de intrare ar avea tensiunea de +4 V, dioda respectivă ar fi în conducție, astfel că la borna de ieșire s-ar obține tensiunea de +4 V. Tensiunea de la borna de ieșire este 0 numai dacă la toate bornele de intrare tensiunea este 0.

Circuitul din figura I.52 realizează funcția de complementare (circuit logic NU). Conform principiilor de funcționare ale unui tranzistor, dacă tensiunea de intrare este +4 V, tranzistorul se află în stare de conducție, astfel că tensiunea de ieșire este 0 V, iar dacă tensiunea de intrare are valoarea de 0 V, tranzistorul este blocat, astfel că tensiunea de ieșire este de +4 V.

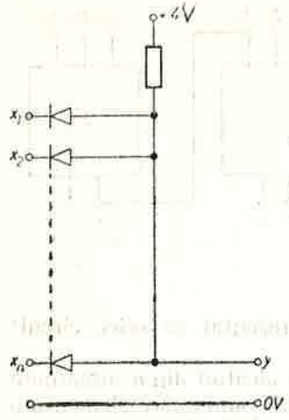


Fig. I.50

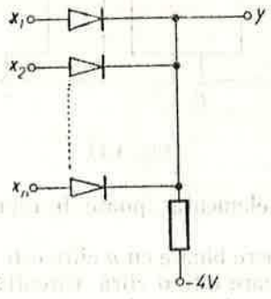


Fig. I.51

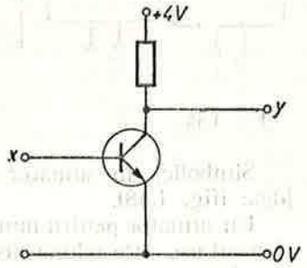


Fig. I.52

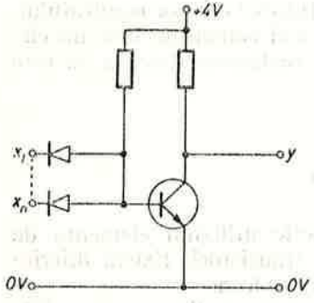


Fig. I.53

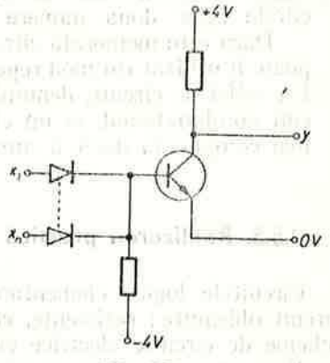


Fig. I.54

Se pot realiza cu uşurinţă circuite logice universale cuplînd un circuit NU cu un circuit SAU şi respectiv cu un circuit ŞI. Se obţin astfel scheme pentru un circuit NU-ŞI (fig. I.53) şi pentru un circuit NICI (fig. I.54).

Tehnica de calcul modernă foloseşte aşa-numitele circuite integrate pentru realizarea funcţiilor booleene. Circuitele integrate sînt de o mare varietate. De obicei au un număr mare de borne (14, 16 sau 32). Unele circuite integrate realizează funcţii simple, elementare. Însă, datorită numărului mare de borne, un circuit integrat poate conţine mai multe circuite logice elementare independente. Există însă circuite integrate care realizează funcţii complexe, cum ar fi codificatoare sau decodificatoare. De asemenea, există circuite integrate care au funcţiunile unor circuite secvenţiale, avînd şi posibilitatea de a memora valoarea unor semnale.

I.5.6. Exerciţii

1. Să se găsească expresiile funcţiilor realizate de circuitele cu contacte din figura I.55, să se simplifice şi să se traseze schemele circuitelor care realizează aceleaşi funcţii, conform expresiilor simplificate.
2. Să se afle expresiile funcţiilor realizate de circuitele din figura I.56.

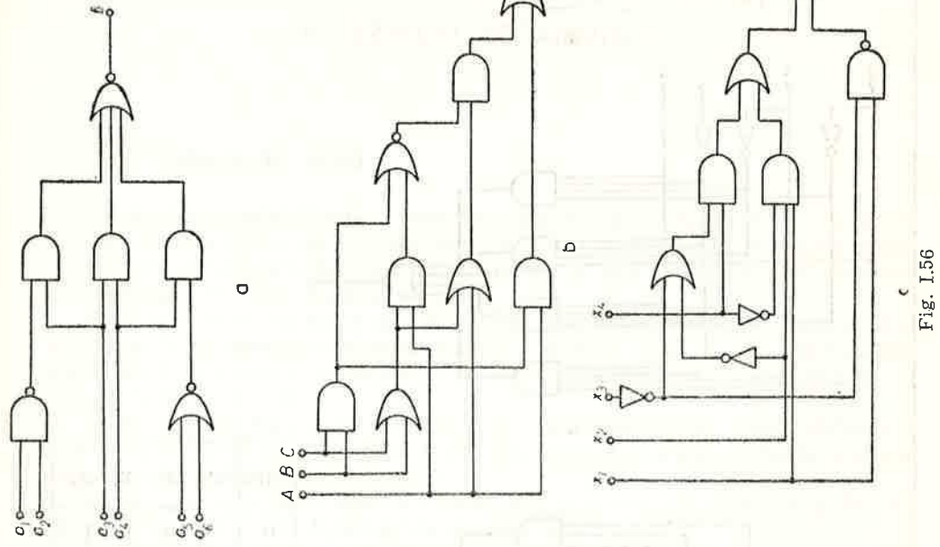


Fig. I.56

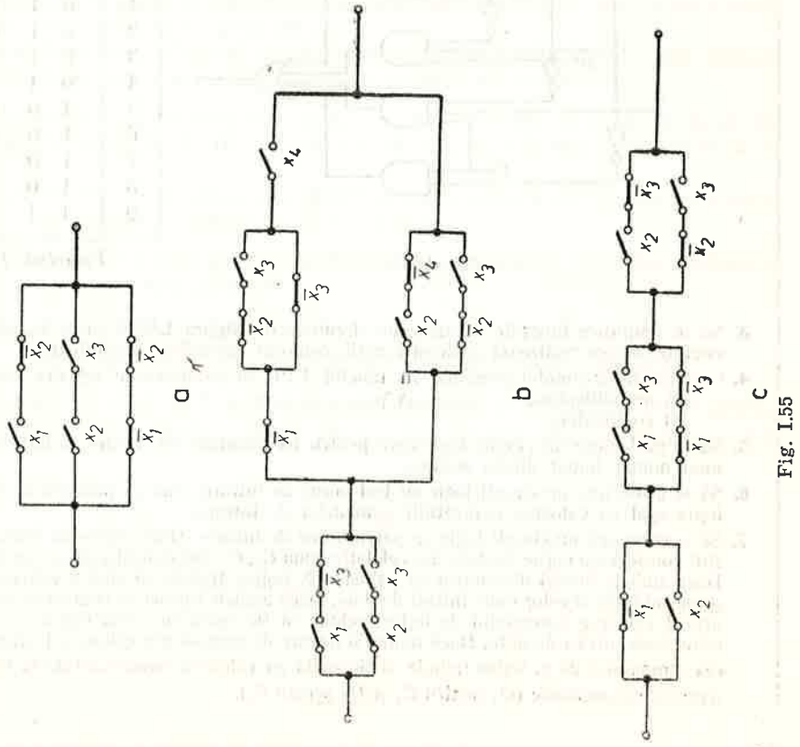
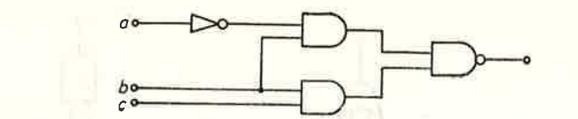
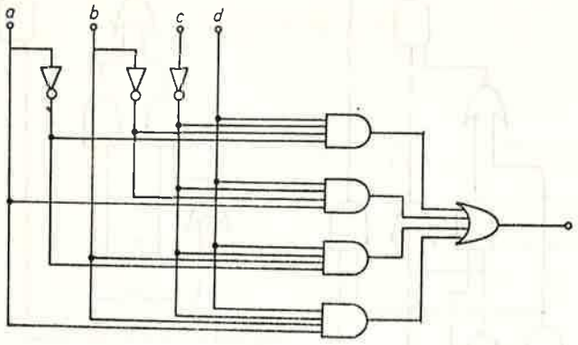


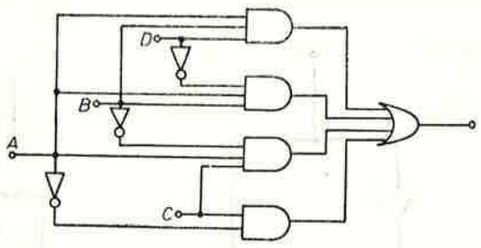
Fig. I.55



a



b



c

Fig. I.57

cifra	x_1	x_2	x_3	x_4
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Tabelul I.19

3. Să se simplifice funcţiile realizate de circuitele din figura I.57 şi să se traseze schema circuitelor care realizează aceleaşi funcţii, conform expresiilor simplificate.
4. Corespunzător codului prezentat în tabelul I.19, să se deseneze schema unui circuit
   a) decodificator ;
   b) codificator.
5. Să se proiecteze un circuit logic care pentru trei semnale de intrare să indice prezenţa unui număr impar dintre acestea.
6. Să se proiecteze un circuit logic cu trei borne de intrare, care să producă un semnal de ieşire egal cu valoarea majorităţii semnalelor de intrare.
7. Să se conceapă un circuit logic cu patru borne de intrare. Două borne de intrare D_1, D_2 sînt considerate borne de date, iar celelalte două C_1, C_2 sînt considerate borne de control. Dacă ambele intrări de control au valoarea 0, ieşirea trebuie să aibă o valoare egală cu produsul logic al celor două intrări de date. Dacă ambele intrări de control au valoarea 1, atunci valoarea semnalului de ieşire trebuie să fie egală cu suma logică a semnalelor celor două intrări de date. Dacă numai o intrare de control are valoarea 1, atunci valoarea semnalului de la ieşire trebuie să fie egală cu valoarea semnalului de la intrarea de date corespunzătoare (D_1 pentru C_1 şi D_2 pentru C_2).

II. GRAFURI NEORIENTATE

II.1. Noţiuni de bază

Un *graf neorientat* G este o pereche ordonată de mulţimi (X, U), unde X este o mulţime finită, iar U este formată din perechi neordonate de elemente, din X. Putem considera deci că U este o familie de submulţimi cu două elemente din mulţimea X. Vom nota $G = (X, U)$.

Mulţimea X se numeşte *mulţimea vîrfurilor* sau a *nodurilor* grafului G şi mulţimea U se numeşte *mulţimea muchiilor* grafului G. O muchie fiind un element din U, ea este o submulţime cu două elemente din X, deci are forma $\{x, y\}$, unde $x, y \in X$.

Vom nota muchia $\{x, y\}$ prin $[x, y]$ şi vom spune că ea uneşte vîrfurile x şi y. Deci notaţiile $[x, y]$ şi $[y, x]$ reprezintă aceeaşi muchie ; vîrfurile x şi y se numesc *extremităţile* acestei muchii.

Dacă $[x, y] \in U$ vom spune că vîrfurile x şi y sînt *adiacente* în graful G, iar vîrfurile x şi y sînt *incidente* cu muchia $[x, y]$.

Deci un graf G poate fi considerat ca o mulţime de vîrfuri, dintre care unele sînt unite două cîte două prin muchii.

Un graf G poate fi desenat în plan reprezentînd vîrfurile sale prin puncte şi muchiile prin linii care unesc anumite perechi de vîrfuri.

Astfel graful $G = (X, U)$, unde $X = \{1, 2, \ldots, 14\}$ şi $U = \{[1, 2], [1, 3], [2, 4], [3, 4], [4, 5], [5, 6], [5, 7], [5, 8], [9, 14], [10, 14], [11, 14], [12, 14], [13, 14]\}$, se reprezintă ca în figura II.1.

De exemplu vîrful 4 este adiacent cu vîrfurile 2, 3 şi 5, vîrful 8 este adiacent cu vîrful 5 iar vîrful 14 este adiacent cu vîrfurile 9, 10, 11, 12 şi 13.

Gradul unui vîrf x este egal, prin definiţie, cu numărul muchiilor incidente cu vîrful x şi se notează cu $d(x)$.

De exemplu, pentru graful din figura II.1 obţinem :

$$d(1) = d(2) = d(3) = 2\,;\ d(4) = 3\,;\ d(5) = 4\,;\ d(6) = d(7) = d(8) = 1.$$

Un vîrf cu gradul egal cu 1 se numeşte *vîrf terminal* al grafului. Pentru graful din figura II.1, vîrfurile 6, 7, 8, 9, 10, 11, 12, 13 sînt vîrfuri terminale. Un

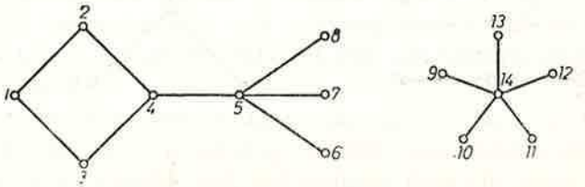


Fig. II.1

vîrf care are gradul egal cu zero, deci care nu mai este adiacent cu nici un alt vîrf al grafului, se numește *vîrf izolat*. Graful din figura II.1 nu conține vîrfuri izolate.

Există o relație simplă între suma gradelor vîrfurilor unui graf și numărul de muchii, dată de propoziția următoare:

Propoziție. *Dacă graful G are m muchii și vîrfurile $x_1, \ldots, x_n$, există relația:*

$$\sum_{i=1}^{n} d(x_i) = 2m.$$

Demonstrație. Fiecare muchie $[x, y]$ a grafului G are două extremități x și y, ea contribuind cu o unitate și la $d(x)$ și la $d(y)$.

Deci suma gradelor grafului este egală cu dublul numărului de muchii.

În particular, suma gradelor este un număr par. De aici mai rezultă următorul corolar.

Corolar. *Pentru orice graf G numărul vîrfurilor de grad impar este par.*

Demonstrație. Să notăm cu S_1 suma gradelor pare și cu S_2 suma gradelor impare ale vîrfurilor grafului G.

Conform propoziției demonstrate putem scrie $S_1 + S_2 = 2m$. Să presupunem, prin reducere la absurd, că numărul vîrfurilor de grad impar ale lui G este un număr impar. În acest caz S_2 este un număr impar. Însă S_1 este un număr par, deoarece fiecare termen din suma care îl definește pe S_1 este număr par. Am ajuns astfel la o contradicție și anume suma dintre un număr par, S_1 și un număr impar, S_2, este un număr par, egal cu $2m$.

Rezultă că presupunerea făcută nu este adevărată, deci proprietatea este demonstrată.

Pentru graful din figura II.1 există 10 vîrfuri de grad impar și anume: 4, 6, 7, 8, 9, 10, 11, 12, 13, 14.

Un *graf parțial* al unui graf $G = (X, U)$, este un graf $G_1 = (X, V)$ care are aceeași mulțime de vîrfuri cu G, iar $V \subset U$. Deci, un graf parțial al lui G este G însuși sau se obține din G prin suprimarea anumitor muchii ale lui G.

Un *subgraf* al unui graf $G = (X, U)$ este prin definiție un graf $H = (Y, V)$, unde $Y \subset X$ iar muchiile din mulțimea V sînt toate muchiile din U care au ambele extremități în mulțimea de vîrfuri Y.

Deci un subgraf H al unui graf G este graful G însuși sau se obține din G prin suprimarea anumitor vîrfuri și a tuturor muchiilor incidente cu acestea.

Vom spune că subgraful H este *indus* sau *generat* de mulțimea de vîrfuri Y.

Astfel, subgraful grafului G din figura II.1, indus de mulțimea de vîrfuri $Y = \{1, 2, 3, 4, 5, 7\}$ este desenat în figura II.2.

Un graf parțial al grafului din figura II.2, obținut din acesta prin suprimarea muchiilor [1, 3], [3, 4] și [4, 5] este desenat în figura II.3.

Spunem că graful din figura II.3 este un subgraf parțial al grafului din figura II.1.

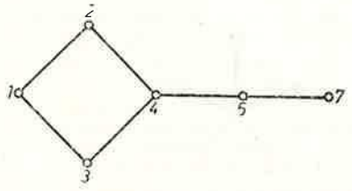


Fig. II.2

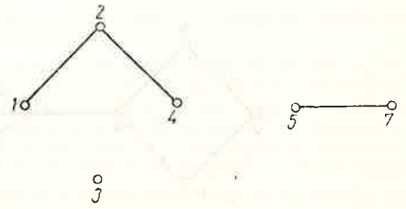


Fig. II.3

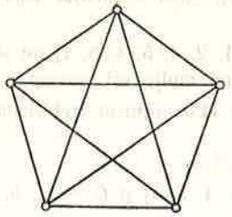


Fig. II.4

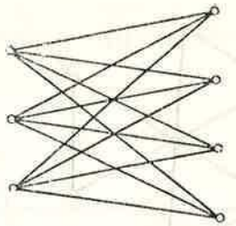


Fig. II.5

Un graf cu n vîrfuri pentru care oricare două vîrfuri sînt adiacente se numește *graf complet* cu n vîrfuri și se notează prin K_n.

În figura II.4 este reprezentat graful K_5.

Deoarece pentru graful K_n oricare două vîrfuri sînt adiacente, rezultă că numărul m de muchii ale acestui graf este egal cu numărul submulțimilor cu 2 elemente ale unei mulțimi cu n elemente, adică $m = C_n^2$.

Un graf G se numește *bipartit* dacă există o partiție a mulțimii vîrfurilor:

$$X = X_1 \cup X_2,\ X_1 \cap X_2 = \varnothing$$

astfel încît fiecare muchie a grafului unește un vîrf din X_1 cu un vîrf din X_2.

Dacă X_1 are p elemente, X_2 are q elemente și oricare vîrf din X_1 este adiacent cu toate vîrfurile din X_2, graful se numește *bipartit complet* și se notează prin $K_{p,q}$. Graful $K_{3,4}$ este desenat în figura II.5.

Deoarece fiecare vîrf $x \in X_1$ are gradul $d(x) = q$, rezultă că numărul de muchii ale grafului $K_{p,q}$ este egal cu pq.

Un *lanț* este un șir (succesiune) de vîrfuri:

$$L = [x_0, x_1, \ldots, x_r]$$

cu proprietatea că oricare două vîrfuri vecine sînt adiacente, adică $[x_0, x_1]$, $[x_1, x_2]$, $\ldots$, $[x_{r-1}, x_r] \in U$. Vîrfurile x_0 și x_r se numesc *extremitățile lanțului* L, iar numărul r se numește *lungimea* acestui lanț. Dacă vîrfurile $x_0, x_1, \ldots, x_r$ sînt distincte două cîte două, lanțul L se numește *elementar*.

Pentru graful din figura II.1 următoarele șiruri de vîrfuri sînt lanțuri:

$$L_1 = [1, 2, 4, 5, 6],\ L_2 = [4, 5, 8, 5, 6],\ L_3 = [9, 14, 10],$$

$$L_4 = [9, 14, 10, 14, 11],\ L_5 = [1, 2, 4, 3, 1].$$

Lanțurile L_1 și L_3 sînt lanțuri elementare, deoarece conțin numai vîrfuri distincte două cîte două. Un lanț $L = [x_0, \ldots, x_r]$ poate fi interpretat ca traseul unei deplasări pe muchiile grafului în ordinea $[x_0, x_1]$, $[x_1, x_2]$, $\ldots$, $[x_{r-1}, x_r]$. De aceea lanțul L de extremități x_0 și x_r se mai spune că este un lanț de la x_0 la x_r sau de la x_r la x_0. Lungimea lanțului L este deci numărul de parcurgeri ale muchiilor grafului G.

Dacă $x_0 = x_r$ și toate muchiile $[x_0, x_1]$, $[x_1, x_2]$, $\ldots$, $[x_{r-1}, x_r]$ sînt distincte două cîte două, lanțul L se numește *ciclu*. Dacă toate vîrfurile ciclului, cu excepția primului și a ultimului vîrf, sînt distincte două cîte două, ciclul se numește *elementar*.

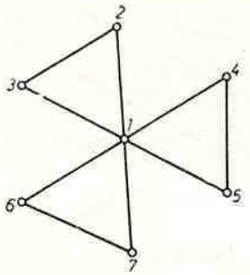


Fig. II.6

Astfel, lanțul L_3 este un ciclu elementar care trece prin vîrfurile 1, 2, 4, 3.

Lanțurile [4, 5, 4] sau [1, 2, 4, 5, 4, 3, 1] nu sînt cicluri deoarece folosesc de mai multe ori aceeași muchie.

Pentru graful din figura II.6 obținem trei cicluri elementare :

$C_1 = [1, 2, 3, 1]$, $C_2 = [1, 4, 5, 1]$ și $C_3 = [1, 6, 7, 1]$.

Deoarece nu contează sensul de deplasare pe fiecare muchie, aceste cicluri pot fi scrise și sub forma :

$C_1 = [1, 3, 2, 1]$, $C_2 = [1, 5, 4, 1]$ și $C_3 = [1, 7, 6, 1]$,

sau alegînd alte vîrfuri ca primele vîrfuri în scrierea ciclului. De exemplu :

$C_1 = [2, 1, 3, 2]$ sau $C_1 = [2, 3, 1, 2]$.

Ciclul $C_4 = [1, 2, 3, 1, 4, 5, 1]$ nu este elementar, deoarece vîrful 1, care este prim și ultim vîrf, se mai repetă o dată în acest șir.

Un graf G se numește *conex* dacă pentru orice pereche de vîrfuri $\{x, y\}$ cu $x \neq y$, există un lanț de la x la y.

Graful G din figura II.1 nu este conex, deoarece nu există nici un lanț între un vîrf din mulțimea $X_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$ și un vîrf din mulțimea $X_2 = \{9, 10, 11, 12, 13, 14\}$. Fiecare din mulțimile X_1 și X_2 induc un subgraf conex al grafului G.

Aceste două subgrafuri conexe se numesc *componentele conexe* ale grafului G.

În general, o componentă conexă C a unui graf G se definește ca fiind un subgraf conex al lui G care este maximal în raport cu această proprietate, adică nu există nici un lanț al lui G care să unească un vîrf din C cu un vîrf care nu aparține lui C.

Pentru graful G din figura II.1 mulțimea de vîrfuri {6, 7, 8} nu induce o componentă conexă deoarece nu induce un subgraf conex. Mulțimea de vîrfuri $Y = \{9, 10, 14\}$ care induce un subgraf conex nu formează o componentă, deoarece nu este maximală în raport cu această proprietate. Într-adevăr, există de exemplu muchia [14, 13] care unește vîrful 14 din Y cu vîrful 13 care nu aparține lui Y.

Graful din figura II.2 este conex, iar graful din figura II.3 are trei componente conexe : una este formată din vîrful izolat 3, alta este formată din vîrfurile 1, 2, 4 și cea de a III-a are vîrfurile 5, 7.

Exemple

1. În figura II.7 este reprezentată o parte a schemei căilor ferate din țara noastră.

Acest desen este un graf, vîrfurile sale reprezentînd nodurile de cale ferată, iar muchiile reprezentînd legăturile directe pe calea ferată dintre două noduri. Dacă cunoaștem distanțele în kilometri asociate fiecărei muchii, ne putem pune problema găsirii celui mai scurt traseu pe calea ferată între două localități.

Acesta va corespunde unui lanț elementar în graful din figura II.7, care unește cele două localități și pentru care suma distanțelor asociate muchiilor este minimă.

O excursie în circuit care trece o singură dată prin anumite localități, întorcîndu-se în localitatea de pornire, va corespunde unui ciclu elementar în acest graf.

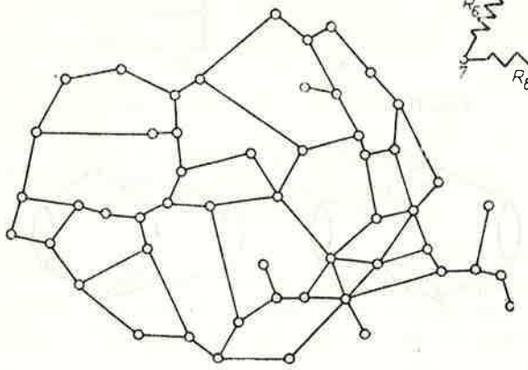


Fig. II.7

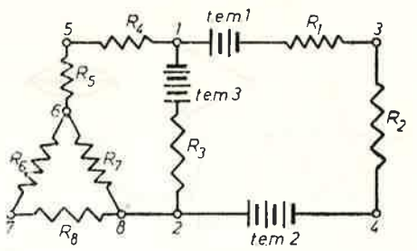


Fig. II.8

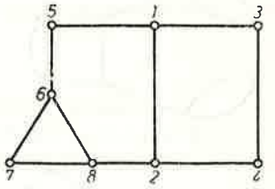


Fig. II.9

2. Vom considera acum un exemplu din fizică și anume calculul intensităților curenților care trec prin ramurile unei rețele electrice, ca cea din figura II.8.

Pentru a rezolva această problemă, cunoscînd schema rețelei, tensiunile electromotoare și valorile rezistențelor, se scriu legile lui Kirchhoff relative la noduri și la ochiuri de rețea.

Făcînd abstracție de elementele de circuit care se găsesc pe laturile schemei putem desena această rețea sub forma grafului din figura II.9. Nodurile rețelei vor corespunde vîrfurilor grafului din figura II.9, iar ochiurile de rețea vor corespunde ciclurilor elementare ale acestui graf.

Fizicianul Kirchhoff a studiat, la mijlocul secolului trecut, rețelele electrice cu metode care aparțin astăzi teoriei grafurilor, contribuind la dezvoltarea acestei teorii.

3. Formulele de structură ale substanțelor chimice sînt grafuri pentru care legăturile dintre vîrfuri corespund legăturilor dintre grupările sau atomii care compun molecula.

Astfel, apa are molecula reprezentată în figura II.10.a, acetilena are formula de structură în figura II.10.b, molecula de benzen este reprezentată în figura II.10.c, iar cea de glucoză în figura II.10.d.

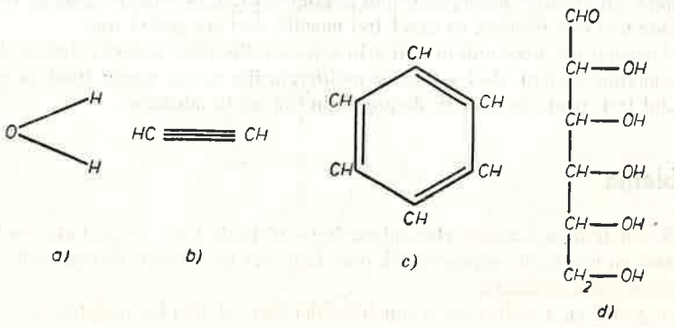


Fig. II.10

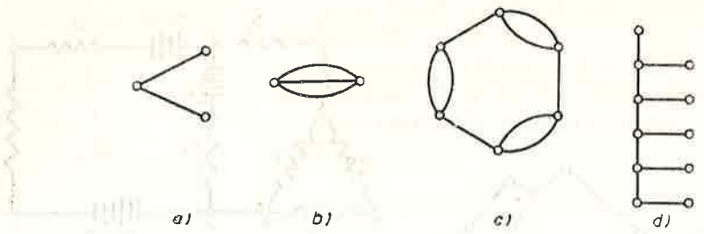


Fig. II.11

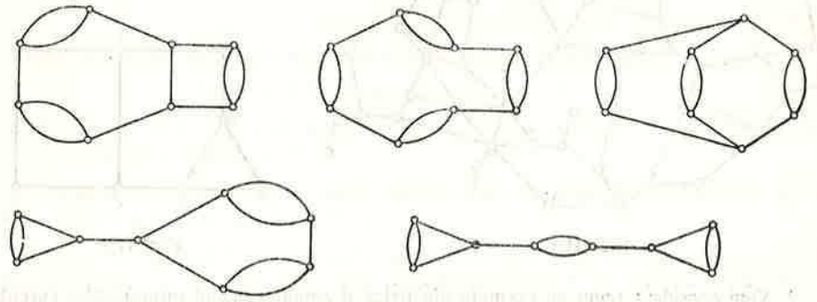


Fig. II.12

În figurile II.11.a – II.11.d aceste formule de structură au fost reprezentate sub formă de grafuri pentru care vîrfurile sînt atomii (respectiv grupările) din moleculă, muchiile reprezentînd legăturile lor chimice.

Grafurile din figurile II.11.b și II.11.c nu sînt grafuri în sensul definiției date, deoarece între anumite perechi de vîrfuri există mai multe muchii.

Un astfel de graf cu muchii multiple se numește *multigraf*.

Într-un graf sau multigraf care este desenul moleculei unei substanțe, gradul unui vîrf este tocmai valența atomului (grupării) respective.

În figura II.12 sînt reprezentați cinci izomeri ai compusului organic numit ciclooctatetrenă, care are formula C_8H_8. Fiecare din cele cinci multigrafuri conține cîte trei muchii duble și șase muchii simple, în vîrfurile acestora găsindu-se gruparea CH, de valență egală cu trei. Din acest motiv fiecare vîrf este incident cu exact trei muchii, deci are gradul trei.

Acești izomeri pot trece unii în alții prin acțiunea diferiților factori exteriori și enumerarea tuturor izomerilor posibili, deci a tuturor multigrafurilor cu un număr fixat de vîrfuri, toate avînd gradul trei, poate da sugestii despre obținerea lor în laborator.

Probleme

II.1.1. Să se determine lanțurile elementare dintre vîrfurile 1 și 7 ale grafului din figura II.13

II.1.2. Dacă un graf G nu conține cicluri, orice lanț care nu folosește de mai multe ori o aceeași muchie este elementar.

II.1.3. Un graf G cu n vîrfuri are m muchii astfel încît să aibă loc inegalitatea:

$$m \geq C_{n-1}^2$$

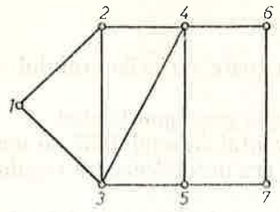


Fig. II.13

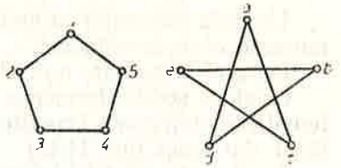


Fig. II.14

Să se arate că G nu are vîrfuri izolate.

II.1.4. Un graf G are n vîrfuri și p componente conexe. Să se arate că numărul m al muchiilor grafului G verifică inegalitatea:

$$m \leq C_{n-p+1}^2.$$

Să se indice pentru ce grafuri această inegalitate devine egalitate.

II.1.5. Să se arate că orice graf conține cel puțin două vîrfuri care au același grad.

II.1.6. Două grafuri $G = (X, U)$ și $H = (Y, V)$ se numesc *izomorfe* dacă există o bijecție:

$$f : X \to Y$$

astfel încît $[x, y] \in U$ dacă și numai dacă $[f(x), f(y)] \in V$. Deci două grafuri izomorfe au același număr de vîrfuri și se obțin unul din celălalt printr-o renumerotare a vîrfurilor. Să se arate că grafurile din figura II.14 sînt izomorfe. Aceeași problemă pentru grafurile din figura II.15.

II.1.7. Fiind dat un graf $G = (X, U)$, *complementarul* său $\bar{G} = (X, \bar{U})$ se definește ca fiind graful cu aceeași mulțime X de vîrfuri, două vîrfuri fiind adiacente în $\bar{G}$ dacă și numai dacă ele nu sînt adiacente în G.

De exemplu unul din cele două grafuri din figura II.14 este complementarul celuilalt.

Să se arate că dacă G nu este conex, atunci complementarul său $\bar{G}$ este conex.

II.1.8. Să se calculeze numărul grafurilor cu n vîrfuri date: $x_1, x_2, \ldots, x_n$.

II.1.9. Pe mulțimea X a vîrfurilor unui graf $G = (X, U)$ se introduce următoarea relație binară:

Spunem că x este în relație cu y și scriem $x \sim y$ dacă $x = y$ sau există un lanț de extremități x și y. Să se arate că această relație binară este o relație de echivalență și clasele acestei echivalențe sînt mulțimile de vîrfuri ale componentelor conexe ale grafului.

II.1.10. Fie G un graf care nu conține cicluri elementare de lungime pară. Să se arate că G conține un vîrf x de grad $d(x) \leq 2$.

II.1.11. Să se găsească toate subgrafurile complete cu 3 vîrfuri ale grafului din figura II.16.

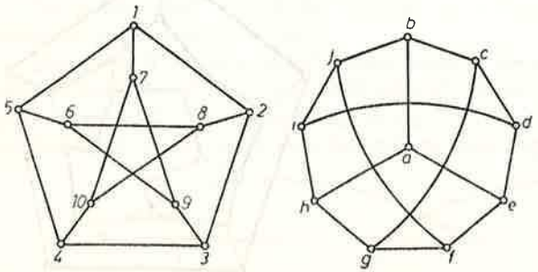


Fig. II.15

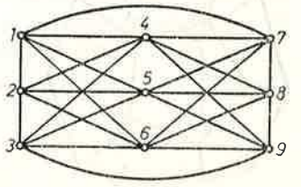


Fig. II.16

II.2. Grafuri hamiltoniene

Un ciclu elementar al unui graf G care trece prin toate vîrfurile grafului se numeşte *ciclu hamiltonian.*

Un graf G care are un ciclu hamiltonian se numeşte *graf hamiltonian.*

Originea acestui termen se găseşte într-un joc inventat în anul 1857 de matematicianul William Hamilton. Partea sa principală era un dodecaedru regulat făcut din lemn (fig. II.17).

Acesta este un poliedru cu 12 feţe care sînt toate pentagoane regulate, iar în fiecare din cele 20 de vîrfuri se întîlnesc cîte 3 muchii. Fiecare vîrf al dodecaedrului lui Hamilton era marcat cu numele unui oraş. Jocul consta în găsirea unui drum de-a lungul muchiilor dodecaedrului care să treacă prin fiecare din cele 20 de oraşe exact o dată şi să se întoarcă în oraşul din care a plecat.

Pentru a uşura memorarea trecerilor efectuate, în fiecare vîrf al dodecaedrului era cîte un cui cu o floare mare, astfel încît în jurul acestor cuie putea să se întindă un fir care să indice drumul parcurs în această călătorie imaginară în jurul lumii.

Problema revine la găsirea unui ciclu hamiltonian în graful format cu vîrfurile şi muchiile dodecaedrului. Acest graf este hamiltonian, un ciclu hamiltonian în reprezentarea plană a grafului dodecaedrului fiind desenat cu linii îngroşate în figura II.18.

O problemă mai generală este aceea a voiajorului comercial, care are următorul enunţ: Un voiajor comercial trebuie să prezinte în n oraşe produsele fabricii pe care o reprezintă, după care se întoarce în oraşul din care a plecat. Cunoscîndu-se costul deplasării între oricare două dintre cele n oraşe, se cere să se determine un traseu care să viziteze o singură dată cele n oraşe şi care să aibă un cost total minim.

În termenii teoriei grafurilor, problema revine la determinarea unui ciclu hamiltonian în graful complet K_n ale cărui vîrfuri reprezintă cele n oraşe, pentru care suma costurilor asociate celor n muchii ale ciclului să fie minimă.

Pentru a rezolva practic această problemă ar trebui definit un algoritm eficient de rezolvare.

Deşi această problemă a fost mult studiată, nu a fost descoperit pînă în prezent un algoritm eficient de rezolvare şi nici nu se cunoaşte dacă poate exista un astfel de algoritm pentru găsirea unui ciclu hamiltonian de cost minim.

Ciclurile hamiltoniene în grafuri particulare au fost de fapt studiate cu mult înainte ca Hamilton să fi propus jocul său.

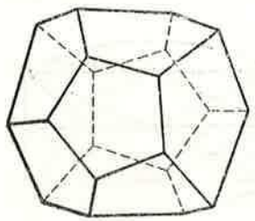


Fig. II.17

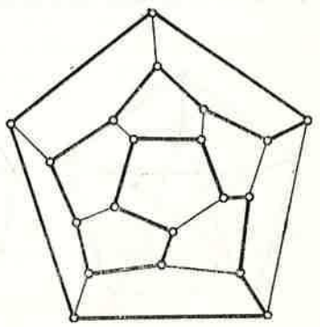


Fig. II.18

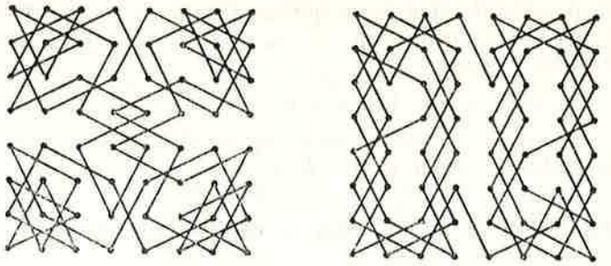


Fig. II.19

În special jocul calului pe o tablă de şah a fost analizat de Euler în anul 1759.

Acest joc cere să se găsească un ciclu hamiltonian în graful cu 64 de vîrfuri care reprezintă cele 64 de pătrate ale unei table de şah şi pentru care două vîrfuri sînt adiacente dacă un cal de pe tabla de şah poate sări în L de pe un pătrat pe celălalt. În figura II.19 sînt reprezentate două soluţii ale acestui joc.

Teorema următoare ne dă o condiţie suficientă pentru existenţa unui ciclu hamiltonian.

Teoremă. *Dacă G este un graf cu $n \geq 3$ vîrfuri astfel încît gradul oricărui vîrf x verifică inegalitatea:*

$$d(x) \geq \frac{n}{2},$$

rezultă că G este hamiltonian.

Demonstraţie. Vom raţiona prin reducere la absurd. Să presupunem deci că pentru orice vîrf x avem $d(x) \geq \frac{n}{2}$ şi G nu conţine nici un ciclu hamiltonian. Vom adăuga muchii între perechi de vîrfuri neadiacente atît timp cît aceasta este posibil, fără ca în graful astfel obţinut să existe un ciclu hamiltonian. Se obţine astfel un graf H cu proprietatea că $d(x) \geq \frac{n}{2}$ pentru orice vîrf x, deoarece prin adăugarea de muchii gradele vîrfurilor cresc. În plus, pentru orice pereche de vîrfuri neadiacente x şi y, prin adăugarea muchiei $[x, y]$ se creează un ciclu hamiltonian care foloseşte muchia $[x, y]$. Deci în graful H există un lanţ elementar de extremităţi x şi y:

$$L = [x_1, x_2, \ldots, x_n],$$

unde $x_1 = x$, $x_n = y$ şi vîrfurile $x_1, \ldots, x_n$ sînt toate cele n vîrfuri ale grafului H.

Este clar că există măcar o pereche de vîrfuri neadiacente x, y în graful H, deoarece în caz contrar $H = K_n$ şi graful complet cu n vîrfuri este hamiltonian.

Să notăm $d(x) = k$ şi fie $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ unde $i_1 = 2 < i_2 < \ldots < i_k$, vîrfurile adiacente cu vîrful x.

Deoarece H nu este hamiltonian, rezultă că y nu este adiacent cu nici unul din vîrfurile $x_{i_1-1}, \ldots, x_{i_k-1}$. Într-adevăr, în caz contrar rezultă că x este adiacent cu un vîrf x_i şi y este adiacent cu x_{i-1}, ceea ce ar produce un ciclu hamiltonian în H. Acest ciclu este desenat cu linie îngroşată în figura II.20.

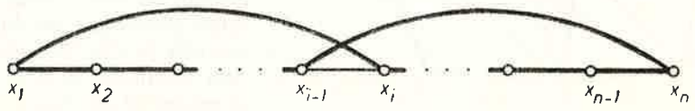


Fig. II.20

Însă această proprietate contrazice ipoteza făcută că H nu este hamiltonian. Deci dintre cele $n-1$ vîrfuri : $x_1, \ldots, x_{n-1}$, vîrful $x_n = y$ nu este adiacent cu cel puțin k vîrfuri, adică :

$$d(y) \leqslant n-1-k \leqslant n-1-\frac{n}{2} = \frac{n}{2} - 1,$$

deoarece în graful H avem $d(x) = k \geqslant \frac{n}{2}$.

Am obținut deci $d(y) < \frac{n}{2}$, ceea ce contrazice proprietatea $d(y) \geqslant \frac{n}{2}$, valabilă pentru orice vîrf din H. Deci am demonstrat prin reducere la absurd că H este hamiltonian.

Probleme

II.2.1. Conține unul din cele două grafuri din figura II.15 un ciclu hamiltonian ? Dar graful din figura II.21 ?

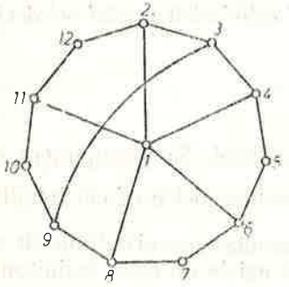


Fig. II.21

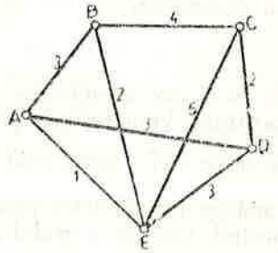


Fig. II.22

II.2.2. Un delegat al unei întreprinderi din orașul A trebuie să meargă în orașele B, C, D și E și apoi să se întoarcă în orașul A.
Cunoscînd costurile deplasării dintr-un oraș în altul, care sînt numerele asociate muchiilor grafului din figura II.22, să se determine succesiunea de orașe pentru care costul total al deplasării să fie minim, știind că sosirea în orașul B este mai urgentă decît sosirea în orașul D.

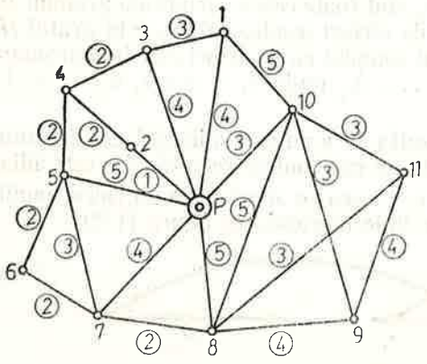


Fig. II.23

II.2.3. O mașină poștală trebuie să plece de la oficiul poștal P, să videze 11 cutii poștale și să se reîntoarcă la oficiul P. Cele 11 cutii sînt reprezentate ca vîrfuri ale grafului din figura II.23, iar distanțele dintre cutii sînt reprezentate prin numere în cerculețe asociate muchiilor acestui graf, care indică posibilitățile de deplasare de la o cutie la alta. Să se găsească un ciclu hamiltonian în acest graf, care să corespundă unui traseu al mașinii poștale de lungime totală minimă.

II.2.4. Să se arate că numărul ciclurilor hamiltoniene ale grafului complet K_n cu $n \geqslant 3$ vîrfuri este egal cu

$$\frac{(n-1)!}{2}.$$

II.2.5. Să se arate că numărul ciclurilor elementare ale grafului complet K_n este egal cu :

$$\frac{1}{2}\sum_{k=3}^{n} \frac{n(n-1)\ldots(n-k+1)}{k}$$

pentru orice $n \geqslant 3$.

II.3. Grafuri euleriene

Un ciclu al unui graf G care conține toate muchiile lui G se numește *ciclu eulerian*. Un graf G care are un ciclu eulerian se numește *graf eulerian*.

Din definiția unui ciclu toate muchiile pe care acesta le conține sînt distincte două cîte două. Deci putem spune că un ciclu hamiltonian trece o singură dată prin toate vîrfurile unui graf, în timp ce un ciclu eulerian trece o singură dată prin toate muchiile unui graf.

Ciclurile euleriene își trag denumirea de la numele matematicianului Leonard Euler care în anul 1736 a caracterizat grafurile care au un astfel de ciclu. Euler a fost condus la această problemă de jocul celor 7 poduri din orașul Kaliningrad. Cele 7 poduri sînt desenate în figura II.24 și problema era următoarea :

Se poate realiza o plimbare peste toate cele 7 poduri, trecînd o singură dată peste fiecare pod ?

Am reprezentat cele 4 regiuni A, B, C, D și cele 7 poduri a, b, c, d, e, f, g ca vîrfuri ale grafului din figura II.25, muchiile grafului reprezentînd posibilitățile de trecere de pe un mal pe un pod sau reciproc.

Problema celor 7 poduri are o soluție dacă există un ciclu eulerian pentru graful din figura II.25. Un astfel de ciclu la fiecare trecere printr-un vîrf utilizează două muchii, care nu mai pot fi folosite pentru o nouă trecere. Existența unui ciclu eulerian pentru acest graf este imposibilă, deoarece, de exemplu, gradul vîrfului D este egal cu 3.

La o primă trecere prin D sînt utilizate două din cele trei muchii, o nouă trecere fiind deci imposibilă și una din muchiile incidente cu D rămîne nefolosită. Această observație simplă ne conduce la următoarea caracterizare a grafurilor euleriene :

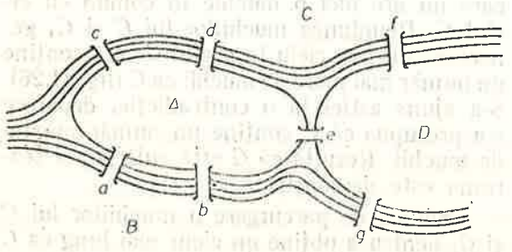


Fig. II.24

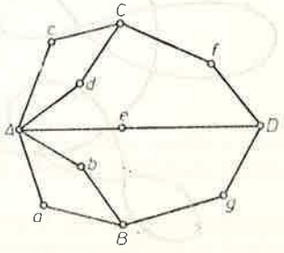


Fig. II.25

Teoremă. *Un graf G fără vîrfuri izolate este eulerian dacă și numai dacă este conex și gradele tuturor vîrfurilor sale sînt numere pare.*

Demonstrație. Să presupunem că graful G nu are vîrfuri izolate și conține un ciclu eulerian. Fie x, y două vîrfuri oarecare ale lui G. Dacă x și y sînt adiacente, rezultă că există un lanț de lungime egală cu unu între x și y. În caz contrar, deoarece G nu are vîrfuri izolate, există două muchii u și v astfel încît u este incidentă cu x și v este incidentă cu y. G fiind eulerian, există un ciclu care trece prin u și v, deci prin vîrfurile x și y. Rezultă că x și y sînt unite printr-un lanț, deci G este conex.

Dacă ciclul eulerian este $[x_1, x_2, \ldots, x_m, x_1]$ și dacă vîrful x apare de k ori în șirul $x_1, x_2, \ldots, x_m$, rezultă $d(x) = 2k$, deoarece fiecare trecere printr-un vîrf utilizează două muchii. Deci toate gradele vîrfurilor lui G sînt numere pare.

Pentru a demonstra suficiența condiției, să presupunem că graful G nu are vîrfuri izolate, este conex și are gradele tuturor vîrfurilor numere pare. Să arătăm că G conține un ciclu eulerian. Să presupunem, prin reducere la absurd, că G nu conține un ciclu eulerian. Deci sau G nu conține cicluri, sau conține numai cicluri care nu parcurg toate muchiile lui G. Fie C un ciclu al grafului G care conține un număr maxim de muchii ale lui G. Vom arăta că presupunerea că C nu există sau C nu conține toate muchiile lui G ne conduce la o contradicție.

Pentru aceasta vom considera graful parțial H al lui G obținut din G prin suprimarea tuturor muchiilor parcurse de ciclul C. Deoarece ciclul C folosește în fiecare vîrf x al lui G un număr par de muchii și conform ipotezei gradele vîrfurilor lui G sînt numere pare, rezultă că gradele tuturor vîrfurilor lui H sînt numere pare, ca diferențe de numere pare. Ciclul C nefiind eulerian, rezultă că H are mulțimea muchiilor nevidă și măcar una din muchiile lui H are în comun una din extremitățile sale, fie z, cu ciclul C. Într-adevăr, în caz contrar ar rezulta că mulțimea vîrfurilor parcurse de ciclul C ar forma o componentă conexă care nu conține toate vîrfurile lui G. Însă acest lucru contrazice ipoteza făcută că G este conex. Să plecăm din vîrful z pe muchia lui H incidentă cu z, deplasîndu-ne pe muchiile grafului H, fără a trece de două ori pe aceeași muchie. Dacă G nu conține cicluri, obținem $H = G$ și alegem pe z un vîrf oarecare al lui G. După un număr finit de astfel de deplasări ne vom întoarce în z. Pentru a justifica această afirmație, să observăm că gradele grafului H fiind numere pare, iar fiecare trecere printr-un vîrf utilizînd exact două muchii care nu mai pot fi parcurse, rămîn tot un număr par de muchii. Deci odată ajunși în oricare vîrf diferit de z al lui H mai rămîn un număr impar, deci nenul, de muchii neutilizate și putem părăsi acel vîrf. Deoarece H are un număr finit de muchii, rezultă că după un număr de pași ne reîntoarcem în s.

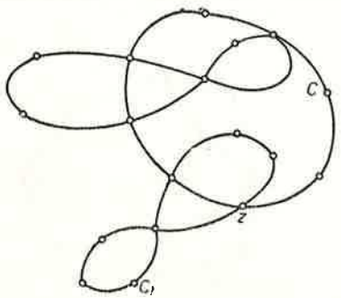


Fig. II.26

Am obținut astfel un ciclu C_1 în graful H, care nu are nici o muchie în comun cu ciclul C. Reuniunea muchiilor lui C și C_1 generează un nou ciclu în graful G care conține un număr mai mare de muchii ca C (fig. II.26). S-a ajuns astfel la o contradicție, deoarece s-a presupus că C conține un număr maxim de muchii. Rezultă că G este eulerian și teorema este demonstrată.

Ordinea de parcurgere a muchiilor lui C și C_1 pentru a obține un ciclu mai lung ca C este de exemplu următoarea:

Plecăm din z și ne întoarcem în z pe muchiile ciclului C, apoi facem aceeași operație pe muchiile lui C_1. Procedeul constructiv utilizat pentru a demonstra suficiența teoremei constituie în același timp un algoritm pentru a obține cicluri din ce în ce mai lungi, pînă la obținerea unui ciclu eulerian într-un graf conex și cu gradele vîrfurilor pare. Ciclurile euleriene intervin într-o serie de probleme de colectare și distribuire. Condiția impusă unui ciclu de a nu parcurge de mai multe ori o aceeași muchie se poate traduce printr-o condiție de lungime minimă a traseului.

Astfel, un poștaș care pleacă de la un oficiu poștal pentru a distribui corespondența pe un număr de străzi și se întoarce la oficiu, trebuie să parcurgă un ciclu eulerian în graful care reprezintă rețeaua stradală respectivă, pentru a avea de străbătut un traseu de lungime minimă.

Probleme

II.3.1. Să se găsească un ciclu eulerian pentru graful din figura II.27.

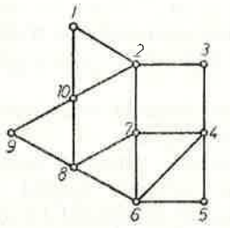


Fig. II.27

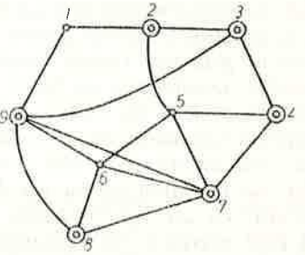


Fig. II.28

II.3.2. Să se indice care este numărul minim de muchii care trebuie adăugate grafului conex din figura II.28 pentru a-l transforma într-un graf eulerian.

II.3.3. Să se indice cum poate fi transformat graful neconex din figura II.29 într-un graf eulerian prin adăugarea unui număr minim de muchii.

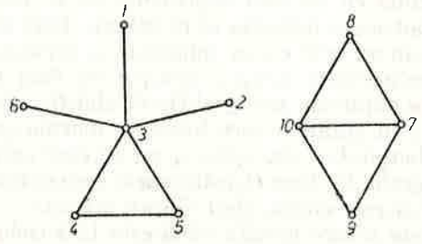


Fig. II.29

II.3.4. Un graf G conține o mulțime de cicluri elementare astfel încît fiecare muchie a lui G aparține exact unuia din aceste cicluri elementare dacă și numai dacă toate gradele vîrfurilor lui G sînt numere pare.

II.3.5. Fie G un graf conex și x și y două vîrfuri distincte ale lui G. Să se arate că există un lanț de extremități x și y care utilizează o singură dată toate muchiile lui G, dacă și numai dacă x și y sînt singurele vîrfuri de grad impar ale grafului.

II.4. Arbori

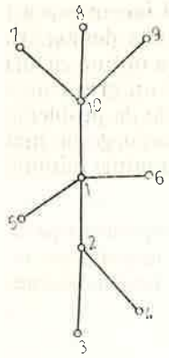


Fig. II.30

Un graf conex și fără cicluri se numește *arbore*. În figura II.30 este desenat un arbore cu 10 vîrfuri.

Se observă din această figură că oricum am suprima o muchie a arborelui se obține un graf neconex care are două componente conexe. De asemenea, oricum am uni printr-o muchie două vîrfuri neadiacente ale unui arbore se creează un ciclu unic. De exemplu, dacă adăugăm muchia [3, 4] apare ciclul [2, 3, 4, 2], dacă adăugăm muchia [5, 7] apare ciclul [5, 1, 10, 7, 5] etc. Aceste proprietăți au loc pentru orice arbore, așa cum rezultă din teorema următoare.

Teoremă. *Următoarele afirmații sînt echivalente pentru un graf G:*

1) *G este un arbore.*

2) *G este un graf conex minimal, adică G este conex și dacă îi suprimăm o muchie oarecare [x, y] graful obținut devine neconex.*

3) *G este un graf fără cicluri maximal, adică G nu conține cicluri și dacă x și y sînt două vîrfuri neadiacente ale lui G, atunci graful obținut din G prin adăugarea muchiei [x, y] conține un ciclu.*

Demonstrație. Vom arăta că 1) $\Rightarrow$ 2), 2) $\Rightarrow$ 1), 1) $\Rightarrow$ 3) și 3) $\Rightarrow$ 1), ceea ce va demonstra echivalența celor trei proprietăți ale unui graf.

1) $\Rightarrow$ 2). Să presupunem că are loc 1), adică G este conex și fără cicluri. Trebuie arătat că are loc 2), adică prin suprimarea oricărei muchii $[x, y]$ se obține un graf neconex. Să presupunem prin reducere la absurd că graful G_1 obținut din G prin suprimarea muchiei $[x, y]$ este conex, deci există un lanț $L = [x_1, \ldots, y]$ de extremități x și y. Dacă se repetă un vîrf z în șirul care îl definește pe L, adică

$$L = [x, \ldots, z, t_1, t_2, \ldots, t_k, z, \ldots, y]$$

vom suprima din lanțul L una dintre aparițiile lui z împreună cu vîrfurile $t_1, \ldots, t_k$ obținînd un nou lanț L' de extremități x și y. Continuînd acest procedeu vom obține în final un lanț elementar de extremități x și y în graful G_1. Acest lanț elementar împreună cu muchia suprimată $[x, y]$ formează un ciclu în arborele G, ceea ce contrazice definiția unui arbore. Deci are loc 2).

2) $\Rightarrow$ 1). Dacă G este un graf conex minimal, să presupunem prin reducere la absurd că G conține un ciclu $[x, z_1, \ldots, z_k, y, x]$. Prin suprimarea muchiei $[x, y]$ a acestui ciclu se obține un nou graf G_1. Graful G_1 este conex deoarece în orice lanț de la u la v în graful G care folosește muchia $[x, y]$ putem înlocui această muchie prin lanțul $L = [x, z_1, \ldots, z_k, y]$ care există în G_1, obținînd un lanț de la u la v în graful G_1. Deci G_1 este conex, ceea ce contrazice 2). Rezultă că G este fără cicluri și este conex, deci G este arbore.

1) $\Rightarrow$ 3). Dacă G este arbore rezultă că G este fără cicluri. Fie x și y două vîrfuri neadiacente ale lui G. Am văzut că există un lanț elementar $[x, z_1, z_2, \ldots, z_k, y]$ de extremități x și y. Deci prin adăugarea muchiei $[x, y]$ se obține un nou graf G_1 care conține ciclul $[x, z_1, \ldots, z_k, y, x]$, adică are loc 3).

3) $\Rightarrow$ 1). Fie G un graf fără cicluri maximal. Trebuie arătat că G este conex. Presupunînd că G nu este conex, rezultă existența a două vîrfuri x și y care aparțin unor componente conexe diferite ale lui G. Prin adăugarea muchiei $[x, y]$ se formează un nou graf G_1 care nu poate conține un ciclu $[x, z_1, z_2, \ldots, z_k, y, x]$, deoarece în acest caz G conține lanțul $[x, z_1, \ldots, z_k, y]$, ceea ce contrazice presupunerea că x și y aparțin unor componente conexe diferite ale lui G. Însă acest lucru contrazice proprietatea 3), deci G este conex. Deoarece G este fără cicluri rezultă că G este arbore și are loc 1). Demonstrația este încheiată.

Un graf parțial H al unui graf G cu proprietatea că H este arbore se numește *arbore parțial* al lui G.

Corolar. *Un graf G conține un arbore parțial dacă și numai dacă G este conex.*

Demonstrație. Pentru a demonstra necesitatea, să observăm că deoarece arborele parțial H este conex și G se obține din H prin unirea prin muchii a unor vîrfuri neadiacente din H, rezultă că și G este conex. Să arătăm acum că orice graf conex G conține un arbore parțial H. Dacă G este un graf conex minimal, din teorema precedentă rezultă că G este arbore, deci vom lua $H = G$. În caz contrar, există o muchie $[x, y]$ a lui G cu proprietatea că graful parțial G_1 obținut din G prin suprimarea muchiei $[x, y]$ este conex. Dacă G_1 este un graf conex minimal vom lua $H = G_1$. În caz contrar vom repeta pentru G_1 procedeul de suprimare a unei muchii aplicat lui G ș.a.m.d., pînă la obținerea unui graf conex minimal, care va fi un arbore parțial al lui G.

Procedeul descris are un număr finit de pași deoarece graful G de la care pornim are cel mult C_n^2 muchii dacă el are n vîrfuri, iar la fiecare etapă se suprimă cîte o muchie a grafului parțial obținut în acel moment.

Propoziția II.4.1. *Orice arbore cu $n \geqslant 2$ vîrfuri conține cel puțin 2 vîrfuri terminale (de gradul 1)*

Demonstrație. Să presupunem, prin reducere la absurd, că există un arbore A cu $n \geqslant 2$ vîrfuri care are cel mult un vîrf de gradul 1.

Fie $L = [x, z_1, \ldots, z_k, y]$ un lanț elementar de lungime maximă al lui A, deci care conține un număr maxim de muchii. Cel puțin una dintre extremitățile lui L, fie aceasta y, are gradul $d(y) \geqslant 2$ deoarece A are cel mult un vîrf de gradul 1. Deoarece y este adiacent cu z_k, rezultă că y mai este adiacent cu un vîrf al lui A. Deoarece lanțul L are o lungime maximă, rezultă că y nu poate fi adiacent decît cu unul dintre vîrfurile $x, z_1, \ldots, z_{k-1}$, ceea ce produce un ciclu în graful A. Deoarece A este arbore am ajuns la o contradicție și proprietatea este demonstrată.

Să observăm că arborele din figura II.30 are 10 vîrfuri și $9 = 10 - 1$ muchii. Această proprietate are loc în general, așa cum ne arată propoziția următoare:

Propoziția II.4.2. *Orice arbore cu n vîrfuri are $n - 1$ muchii.*

Demonstrația se face prin inducție după n. Pentru $n = 1$ există un singur arbore cu un vîrf și fără nici o muchie, pentru $n = 2$ obținem de asemenea un arbore unic K_2 cu 2 vîrfuri și o singură muchie.

Să presupunem că proprietatea este adevărată pentru orice arbore cu cel mult n vîrfuri și fie A un arbore cu $n + 1$ vîrfuri și m muchii. Conform propoziției precedente, arborele A are cel puțin două vîrfuri de gradul 1. Fie x unul dintre acestea și A_1 subgraful obținut din A prin suprimarea vîrfului x și a muchiei incidente cu x. Deoarece A nu conține cicluri, rezultă că nici A_1 nu conține cicluri. Vom arăta că în plus graful A_1 este conex, deci A_1 este un arbore cu n vîrfuri. Fie u și v două vîrfuri diferite între ele și diferite de x ale arborelui A. Deoarece A este conex rezultă că există un lanț, deci și un lanț elementar $L = [u \ldots, v]$. Dacă muchia incidentă cu x este $[x, y]$, rezultă că lanțul elementar L nu folosește muchia $[x, y]$ pentru că $d(x) = 1$. Deci L este un lanț de extremități u și v și pentru subgraful A_1. Rezultă că A_1 este conex. Aplicînd ipoteza de inducție pentru A_1 găsim că el are un număr de muchii egal cu $m_1 = n - 1$. Dar A conține în plus față de A_1 muchia $[x, y]$, deci A are $m = (n - 1) + 1 = n$ muchii și proprietatea este demonstrată.

Proprietatea unui arbore de a fi un graf conex minimal face ca arborii să intervină într-o serie de probleme de optimizare, cum este următoarea:

Problema arborelui parțial minim

Să considerăm un graf conex $G = (X, U)$ și o funcție $c : U \to \mathbf{R}_+$ care asociază fiecărei muchii a grafului G un număr real pozitiv numit *costul* acelei muchii.

Costul unui graf parțial $H = (X, V)$ al lui G este egal prin definiție cu suma costurilor asociate muchiilor lui H, ceea ce vom nota prin :

$$c(H) = \sum_{u \in V} c(u).$$

Se pune problema determinării unui graf parțial H al lui G care să fie conex și să aibă un cost minim.

Un astfel de graf parțial de cost minim trebuie să fie un arbore parțial, deoarece arborii sînt singurele grafuri conexe minimale.

Într-adevăr, dacă H este un graf parțial de cost minim al lui G și H conține o muchie $u = [x, y]$ a cărei suprimare conduce la un alt graf parțial H_1, conex, rezultă că :

$$c(H) = c(H_1) + c(u) > c(H_1).$$

Însă inegalitatea obținută $c(H_1) < c(H)$ contrazice minimalitatea grafului H.

Rezultă că orice graf parțial de cost minim care este conex este un arbore parțial al lui G. Dacă vîrfurile grafului G reprezintă de exemplu nodurile unei rețele de telecomunicații, iar costul unei muchii reprezintă costul instalării unei linii telefonice între cele 2 noduri ale rețelei reprezentate de extremitățile muchiei, problema determinării rețelei conexe de cost minim este tocmai problema găsirii unui arbore parțial de cost minim în graful care reprezintă rețeaua.

Rețeaua de comunicații obținută trebuie să fie conexă pentru a facilita realizarea de convorbiri telefonice între oricare două noduri ale rețelei, direct sau indirect, printr-o serie de noduri intermediare.

Pentru găsirea unui arbore parțial de cost minim, pe care îl vom numi în continuare arbore parțial minim al unui graf conex, prezentăm următorul algoritm :

Algoritmul APM (arbore parțial minim)

Fiind dat un graf conex $G = (X, U)$ cu o funcție cost $c : U \to \mathbf{R}_+$, se alege o muchie u cu costul $c(u)$ minim. Dintre muchiile nealese se va selecta mereu muchia de cost minim care nu formează cicluri cu muchiile deja alese.

Aplicarea acestui algoritm se termină cînd se obține o mulțime de muchii V, deci un graf parțial $H = (X, V)$ al lui G cu $V \subset U$, cu proprietatea că oricare dintre muchiile rămase ale lui G formează cicluri cu muchiile lui H. Deci H este un graf fără cicluri, maximal, cu aceeași mulțime de vîrfuri ca G.

Conform teoremei demonstrate rezultă că H este un arbore parțial al lui G.

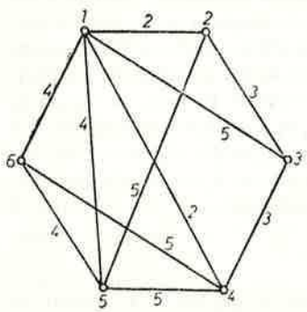


Fig. II.31

Să aplicăm algoritmul *APM* pentru graful conex G cu 6 vîrfuri din figura II.31, pentru care costurile muchiilor sînt desenate lîngă muchiile respective.

Alegem mai întîi o muchie de cost minim, de exemplu [1, 2] cu costul 2, apoi muchia [1, 4] care are de asemenea un cost minim egal cu 2. În continuare există două muchii dintre cele nealese avînd costul minim egal cu 3 unități și anume [2, 3] și [3, 4]. O alegem de exemplu pe [2, 3]. Muchia rămasă de cost minim este [3, 4], dar ea nu mai poate fi aleasă deoarece formează ciclul [1, 2, 3, 4, 1] cu muchiile deja alese. Alegem de exemplu muchia [1, 6] cu costul 4 și apoi muchia [1, 5] cu costul 4, fără a apărea cicluri formate cu muchiile alese. Am obținut 5 muchii : [1, 2], [2, 3], [1, 4], [1, 6], [1, 5], deci am obținut un arbore parțial al grafului G, deoarece propoziția II.4.2 ne arată că un arbore cu n vîrfuri are $n - 1$ muchii. Se observă ușor că oricum am adăuga o nouă muchie grafului parțial obținut în figura II.32 apare un ciclu.

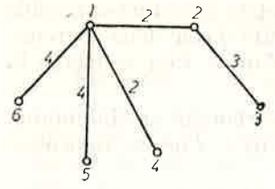


Fig. II.32

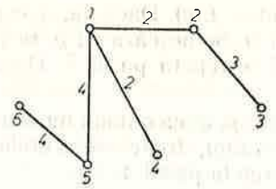


Fig. II.33

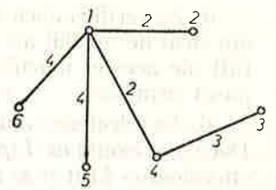


Fig. II.34

Deci graful din figura II.32 este un arbore parțial minim, de cost egal cu 15, al grafului conex din figura II.31.

Dacă în locul muchiei [1, 6] alegem muchia [5, 6] cu același cost egal cu 4, obținem arborele minim din figura II.33. Se mai poate obține un arbore minim înlocuind muchia [1, 5] prin muchia [5, 6] pentru arborele din figura II.32. Se mai obțin trei arbori minimi dacă se înlocuiește, pentru fiecare din cei trei arbori obținuți, muchia [2, 3] prin muchia [3, 4] de același cost, egal cu 3. Unul dintre acești arbori este desenat în figura II.34. Rezultă că graful din figura II.31 are 6 arbori parțiali minimi.

În general, dacă există mai multe muchii cu același cost, pentru un graf conex G pot exista mai multe posibilități de alegere a unei muchii de cost minim care nu formează cicluri cu muchiile deja alese, deci pot exista mai mulți arbori parțiali minimi.

Vom încheia acest paragraf indicînd o formă a algoritmului *APM* care este ușor programabilă pentru un calculator electronic.

Pentru aceasta, să observăm că inițial se pleacă cu un graf parțial al grafului G care nu conține nici o muchie, deci care conține n vîrfuri izolate dacă G are n vîrfuri. Ulterior, prin adăugare de muchii se formează grafuri parțiale care nu conțin cicluri, deci care au drept componente conexe arbori. Se observă că o nouă muchie u poate fi selectată dacă are un cost minim printre muchiile nealese și dacă extremitățile ei aparțin unor componente conexe diferite ale grafului parțial obținut pînă în acel moment.

În caz contrar apare un ciclu, deoarece conform teoremei demonstrate un arbore este un graf fără cicluri, maximal.

Pentru a memora numerele de ordine ale componentelor conexe în care se găsesc la un moment dat vîrfurile grafului G vom folosi o *listă* cu n poziții, astfel încît poziția i din listă, notată cu $L(i)$, să indice numărul de ordine al componentei în care se găsește vîrful i al grafului.

Pentru a ușura căutarea muchiei de cost minim, vom alcătui lista muchiilor grafului G în ordine crescătoare a costurilor. Algoritmul se va opri după ce a selectat exact $n - 1$ muchii, deoarece un arbore cu n vîrfuri are $n - 1$ muchii.

Algoritmul *APM* devine :

1. Pentru $i = 1, \ldots, n$ se face $L(i) \leftarrow i$;
2. Se alcătuiește lista muchiilor grafului G în ordinea crescătoare a costurilor ;
3. Fie $[p, q]$ prima muchie din șirul muchiilor lui G ;
4. Au fost selectate $n - 1$ muchii ? Dacă da, stop.
   Am obținut un arbore parțial minim.
   Dacă nu, mergi la pasul următor.

5. Se verifică dacă $L(p) = L(q)$. Dacă da, se consideră următoarea muchie din șirul de muchii ale lui G. Se notează cu p, respectiv q cele două extremități ale acestei muchii și se repetă pasul 5. Dacă $L(p) \neq L(q)$ se merge la pasul următor.

6. Se selectează muchia $[p, q]$ ca o nouă muchie a arborelui parțial minim. Dacă de exemplu $L(p) < L(q)$, toate elementele $L(i) = L(q)$ se înlocuiesc cu valoarea $L(p)$ și se merge la pasul 4.

Se observă că dacă $L(p) = L(q)$, cele două extremități ale muchiei $[p, q]$ sînt în același arbore, deci alegerea lui $[p, q]$ ar crea un ciclu în graful parțial obținut la momentul respectiv. Deci trebuie considerată următoarea muchie din șirul de muchii. La pasul 6 se unifică componentele conexe cărora le aparțin cele două extremități ale muchiei $[p, q]$,dînd tuturor vîrfurilor din reuniunea celor două componente numărul de ordine egal cu $L(p)$. La sfîrșitul aplicării algoritmului lista L va avea toate pozițiile egale cu 1. De fapt, după selectarea ultimei muchii putem să ne oprim, fără a mai unifica numerele de ordine ale celor două ultime componente conexe.

Pentru exemplul grafului din figura II.31, o ordonare posibilă a muchiilor în ordinea crescătoare a costurilor este următoarea :
[1, 2], [1, 4], [2, 3], [3, 4], [5, 6], [1, 5], [1, 6], [4, 5], [1, 3], [2, 5], [4, 6]. Inițial, lista L are forma | 1 | 2 | 3 | 4 | 5 | 6 |. Deoarece $L(1) \neq L(2)$ se selectează muchia [1, 2] și se obține lista L : | 1 | 1 | 3 | 4 | 5 | 6 |. Acum $L(1) = 1 \neq L(4) = 4$, deci alegem și muchia [1, 4].

Noua listă L : | 1 | 1 | 3 | 1 | 5 | 6 |. Obținem $L(2) = 1 \neq L(3) = 3$, deci alegem și muchia [2, 3], reactualizînd lista L : | 1 | 1 | 1 | 1 | 5 | 6 |. Deoarece $L(3) = L(4) = 1$ nu vom alege muchia [3, 4], trecînd la următoarea muchie [5, 6]. Deducem $L(5) = 5 \neq L(6) = 6$, deci selectăm muchia [5, 6] și găsim noua listă L : | 1 | 1 | 1 | 1 | 5 | 5 |. Selectăm și a V-a muchie [1, 5] deoarece $L(1) = 1 \neq L(5) = 5$ obținînd arborele din figura II.33 și noua listă L cu toate componentele egale cu 1.

Probleme

II.4.1. Să se găsească un arbore parțial minim al grafului conex din figura II.35.

II.4.2. Fie G un graf cu $n \geq 3$ vîrfuri. Să se arate că următoarele afirmații sînt echivalente :
a) G este un arbore ;
b) G este conex și are $n - 1$ muchii ;
c) G este fără cicluri și are $n - 1$ muchii.

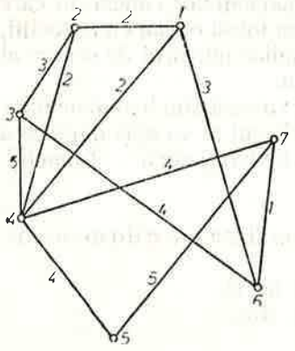


Fig. II.35

II.4.3. Dacă G este un graf conex cu n vîrfuri, să se arate că pentru orice k astfel încît $1 \leq k \leq n$ există un subgraf conex al lui G cu k vîrfuri.

II.4.4. Să se arate că un graf cu n vîrfuri și cel puțin n muchii conține cel puțin un ciclu.

II.4.5. Fie G un graf conex cu n vîrfuri și m muchii. Să se arate că numărul minim de muchii care trebuie înlăturate astfel încît graful obținut să nu conțină cicluri este egal cu $m - n + 1$ (acest număr se numește *numărul ciclomatic* al grafului).

II.4.6. Numerele $d_1 \geq d_2 \geq \ldots \geq d_n \geq 1$ sînt gradele vîrfurilor unui arbore cu $n \geq 2$ vîrfuri, dacă și numai dacă $d_1 + d_2 + \ldots + d_n = 2n - 2$.

II.5. Arbori binari și aplicații

Un *arbore binar* se definește în modul următor :

Este un arbore care are un vîrf numit *rădăcină*, al cărui grad este zero, unu sau doi. Dacă gradul rădăcinii este zero, arborele binar este format numai din rădăcină. În caz contrar, rădăcina se leagă printr-o muchie sau prin două muchii de unul sau două alte noi vîrfuri care se desenează sub rădăcină și care se numesc *fiii* vîrfului rădăcină. Modul în care vîrfurile fiu se desenează sub rădăcină, la stînga sau la dreapta, are importanță. Deci vom face distincție între fiul din dreapta și fiul din stînga rădăcinii. Aceste noduri fiu au fiecare zero, unul sau două noduri fiu, la stînga sau la dreapta ș.a.m.d. Vom spune că rădăcina arborelui are nivelul 0, fiii rădăcinii nivelul 1, fiii acestora nivelul 2, descendenții de ordin k ai rădăcinii nivelul k și îi vom desena la aceeași înălțime față de marginea de jos a unei pagini.

În figura II.36 este reprezentat un arbore binar cu rădăcina A. Pe nivelul 1 apar vîrfurile B și C ; B este fiul din stînga al lui A și C este fiul din dreapta al lui A. Pe nivelul 2 se găsesc vîrfurile D și E, iar pe nivelul 3 vîrfurile F și G.

Vîrfurile terminale sau vîrfurile de grad 1 diferite de rădăcină ale acestui arbore sînt D, F, G, C. Se observă că aceste vîrfuri sînt vîrfurile care nu au fii. În figura II.37 sînt desenați doi arbori binari care sînt identici ca arbori, dar sînt distincți ca arbori binari, deoarece pentru unul dintre ei B este fiul din stînga al rădăcinii A, iar pentru celălalt B este fiul din dreapta al rădăcinii A.

Dacă suprimăm rădăcina și muchiile incidente cu aceasta, arborii obținuți se numesc *subarborele stîng*, respectiv *subarborele drept* al rădăcinii. Unul dintre aceștia sau amîndoi pot fi vizi. De exemplu pentru arborele binar din figura II.36 vîrful A are subarborele stîng format din vîrfurile B, D, E, F, G iar subarborele drept format din vîrful C.

Noțiunea de subarbore se aplică și altui vîrf diferit de rădăcină.

De exemplu vîrful terminal D are ambii subarbori stîng și drept vizi, vîrful E are subarborele stîng format din vîrful F și subarborele drept format din vîrful G.

Dacă fiecare vîrf care nu este vîrf terminal al unui arbore binar are exact doi fii, arborele binar se numește *complet*.

Arborele binar din figura II.36 este complet, dar arborii binari din figura II.37 nu sînt compleți.

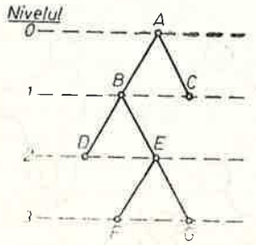


Fig. II.36

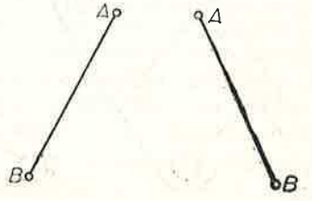


Fig. II.37

II.5.1. **Notația fără paranteze a unei expresii aritmetice**

Oricărei expresii aritmetice formate din constante și variabile și care utilizează operațiile de adunare, scădere, înmulțire, împărțire și ridicare la putere i se poate asocia un arbore binar. De exemplu, expresiei :

$$(x + y)z + 7$$

i se asociază arborele binar complet din figura II.38, unde semnul * reprezintă înmulțirea.

În vîrfurile acestui arbore sînt reprezentați operatorii, adică operațiile din expresia aritmetică sau operanzii, adică constantele și variabilele din expresie. Ultima operație efectuată este adunarea, care se reprezintă în rădăcina arborelui binar. Subarborele stîng, respectiv subarborele drept reprezintă cele două expresii cărora li se aplică această ultimă operație de adunare și anume $(x + y)z$ și 7. Pentru expresia $(x + y)z$ se aplică aceeași regulă de reprezentare, deci în rădăcina subarborelui stîng vom reprezenta operația de înmulțire ș.a.m.d. Constanta 7 se reprezintă în rădăcina subarborelui drept, care nu mai conține alte vîrfuri.

Să reprezentăm după aceeași regulă expresia :

$$(x^2 - 3y)t + \frac{x}{y + 4}.$$

Se obține arborele binar complet din figura II.39, unde semnul / reprezintă operația de împărțire, iar ↑ reprezintă operația de ridicare la putere. Deci vom nota :

$$\frac{a}{b} = a/b, \text{ iar } a^b = a \uparrow b.$$

Un arbore asociat unei expresii aritmetice este un *arbore binar complet*, oricare vîrf neterminal avînd asociat un operator are exact doi fii. Într-adevăr, operațiile întîlnite într-o expresie aritmetică sînt operații binare, adică au doi operanzi. Vîrfurile terminale ale arborilor din figurile II.38 și II.39 au asociați operanzii din expresia aritmetică, care sînt constantele și variabilele expresiei.

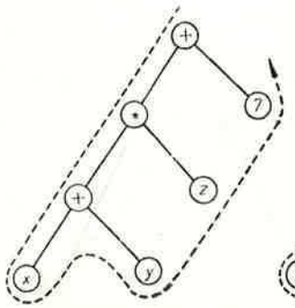


Fig. II.38

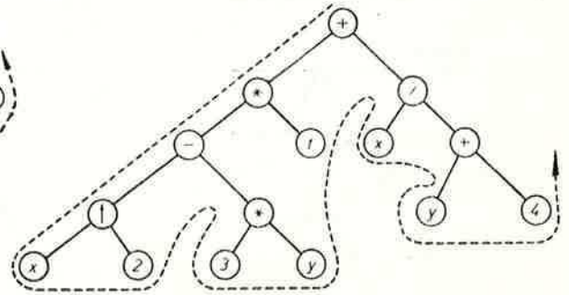


Fig. II.39

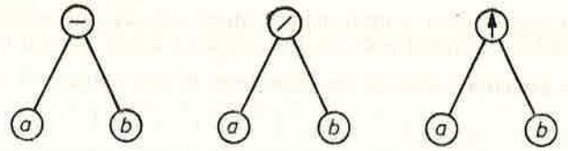


Fig. II.40

Se observă că deoarece operațiile de scădere, împărțire și ridicare la putere nu sînt comutative, arborii obținuți sînt într-adevăr niște arbori binari, pentru care se face distincție între fiul din dreapta și fiul din stînga al fiecărui vîrf neterminal.

Așa cum am făcut în figurile II.38 și II.39, vom reprezenta expresiile $a - b$, $\frac{a}{b}$ și a^b după cum se arată în figura II.40, cu a desenat la stînga și b desenat la dreapta vîrfului care reprezintă operatorul de scădere, împărțire sau ridicare la putere.

În general, pentru a reprezenta o expresie aritmetică E sub forma unui arbore binar, procedăm după regula următoare :

a) Dacă E este o constantă sau o variabilă, arborele binar se reduce la rădăcină, căreia îi asociem constanta sau variabila respectivă ;

b) Dacă $E = E_1 \alpha E_2$, unde α este unul dintre operatorii : +, —, *, /, ↑, vom asocia rădăcinii arborelui binar operatorul α, subarborele stîng va reprezenta după aceeași regulă expresia aritmetică E_1, iar subarborele drept va reprezenta după aceeași regulă expresia E_2.

Această regulă conduce, prin recurență, la un arbore binar complet asociat expresiei E. Invers, fiind dat un arbore binar complet care conține în nodurile terminale constante sau variabile și în celelalte noduri operatori, se poate obține ușor expresia aritmetică asociată. Din punctul de vedere al calculului automat, cu ajutorul unui calculator electronic, o altă formă a unei expresii aritmetice este mai utilă. Această formă se obține de exemplu printr-un procedeu de *parcurgere* a vîrfurilor arborelui, plecînd din rădăcină și trecînd prin fiecare vîrf o singură dată. Parcurgerea arborelui binar asociat unei expresii aritmetice are loc după regula următoare : se pleacă din rădăcină pe muchia din stînga, iar în momentul cînd întreg subarborele stîng al unui vîrf a fost parcurs, se parcurge subarborele drept al acelui vîrf după aceeași regulă. De fiecare dată cînd se trece printr-un vîrf se scrie operatorul sau operandul asociat acelui vîrf.

Pentru arborii binari din figurile II.38 și II.39 ordinea de parcurgere a fost desenată printr-o linie punctată.

De exemplu, pentru arborele din figura II.38 plecăm din rădăcină pe muchia din stînga și ne deplasăm pe muchiile din stînga pînă ajungem în vîrful terminal asociat variabilei x. Deoarece subarborele stîng al vîrfului diferit de rădăcină, asociat cu operația +, a fost parcurs, parcurgem subarborele drept, care se reduce la rădăcină și are asociată variabila y. Acum subarborele stîng al vîrfului asociat cu * a fost parcurs, vom parcurge subarborele drept, care se reduce la rădăcină și are asociată variabila z. Am parcurs subarborele stîng al rădăcinii întregului arbore, deci mai rămîne de parcurs subarborele drept care se reduce la rădăcină și are asociată constanta 7.

Am găsit succesiunea : $+ * + x\ y\ z\ 7$, care se numește *notația poloneză* sau *notația fără paranteze* a expresiei aritmetice $(x + y)z + 7$. Se folosește termenul de notație poloneză pentru șirul de semne obținut în urma par-

curgerii cu regula dată a unui arbore binar asociat unei expresii aritmetice, deoarece matematicianul polonez Lukasiewicz a fost primul care a definit-o.

În urma parcurgerii arborelui binar din figura II.39 se obține șirul de semne :

$$+ \; * \; - \; \uparrow \; x \; 2 \; * \; 3 \; y \; t \; / \; x \; + \; y \; 4,$$

care reprezintă notația poloneză sau notația fără paranteze a expresiei aritmetice :

$$(x^2 - 3y)t + \frac{x}{y+4}.$$

Pentru a calcula valoarea numerică a unei expresii aritmetice, trebuie ca toate variabilele să aibă atribuite anterior anumite valori numerice. Calculul valorii numerice a unei expresii aritmetice cu paranteze, pentru care toate variabilele au atribuite anumite valori numerice, se realizează într-un calculator, după ce a fost obținută notația poloneză a acelei expresii, în modul următor :

Se detectează în șirul de semne care alcătuiesc notația poloneză ultimul operator în sensul de la stînga la dreapta.

Se aplică acel operator celor două valori numerice care îl urmează în șir și numărul astfel obținut este plasat în șir în locul operatorului și a celor două numere care au fost utilizate în calcul.

Șirul nou obținut conține cu 2 semne mai puțin decît vechiul șir. Dacă am obținut un singur număr ne oprim, deoarece el reprezintă valoarea numerică a expresiei aritmetice. Dacă nu, aplicăm același procedeu șirului de semne obținut în acest moment.

În cazul șirului $+ \; * \; + \; x \; y \; z \; 7$ se va înlocui succesiunea $+ \; x \; y$ prin rezultatul adunării valorii lui x cu valoarea lui y pe care îl notăm a. Se obține șirul : $+ \; * \; a \; z \; 7$, unde $a = x + y$. Se înmulțește a cu z și se obține șirul : $+ \; b \; 7$, unde $b = az$. În final se obține o singură valoare, $b + 7 = az + 7 = (x + y)z + 7$, adică valoarea numerică căutată.

Vedem deci cum un studiu de logică matematică și-a găsit aplicații la programarea calculatoarelor electronice, care au apărut cu 15 ani mai tîrziu.

II.5.2. Arbori de sortare

A *sorta* n numere reale $a_1, a_2, \ldots, a_n$ înseamnă a scrie aceste numere în ordine crescătoare, adică a găsi o permutare p a mulțimii $\{1, \ldots, n\}$ care verifică :

$$a_{p(1)} \leqslant a_{p(2)} \leqslant \ldots \leqslant a_{p(n)}.$$

Dacă numerele $a_1, \ldots, a_n$ se află depuse în această ordine în memoria unui calculator, sortarea lor presupune rearanjarea acestor numere astfel încît ele să apară în ordine crescătoare.

Vom vedea cît de ușor putem găsi unul din aceste numere în cazul cînd ele sînt sortate, în raport cu situația cînd numerele sînt aranjate într-o ordine oarecare. Se apreciază că cel puțin o treime din timpul de lucru al calculatoarelor care lucrează în domeniul informaticii de gestiune este afectat unor sortări de date.

Vom presupune mai întîi că vrem să determinăm cel mai mic dintre numerele $a_1, \ldots, a_n$, pe care îl notăm $\min(a_1, \ldots, a_n)$, comparînd numerele două cîte două.

Propoziție. *Numărul minim de comparații a cîte două numere necesare pentru a găsi* $\min(a_1, \ldots, a_n)$ *este egal cu* $n - 1$.

Demonstrație. Să notăm numărul minim de comparații a cîte două numere necesar pentru a găsi minimul din n numere prin $f(n)$.

Dacă $n = 2$ găsim minimul comparînd cele două numere, deci $f(2) = 1$.

Presupunînd proprietatea adevărată pentru orice $m \leqslant n - 1$, fie n numere $a_1, \ldots, a_n$. După prima comparație găsim $b = \min(a_i, a_j)$, unde $1 \leqslant i < j \leqslant n$. Rămîne să determinăm minimul din $n - 1$ numere, care formează mulțimea $\{a_1, \ldots, a_n\} \setminus \{a_i, a_j\}$, la care se adaugă numărul b.

Deci putem scrie :

$$f(n) = 1 + f(n - 1).$$

Deoarece conform ipotezei de inducție $f(n - 1) = n - 2$, rezultă că $f(n) = n - 1$ și proprietatea este demonstrată.

Procesul de găsire a minimului din n numere se poate reprezenta printr-un arbore binar. Într-adevăr, se poate proceda prin analogie cu reprezentarea unei expresii aritmetice printr-un arbore binar, deoarece operația de găsire a minimului din două numere este o operație binară.

De exemplu, dacă determinăm $\min(2, 7, 5, 1)$ putem proceda astfel : găsim $\min(2, 7) = 2$, apoi $\min(2, 5) = 2$ și în fine $\min(2, 1) = 1$.

Acest calcul este reprezentat prin arborele binar complet din figura II.41.

În rădăcina arborelui apare numărul 1, care este rezultatul și fiecare vîrf neterminal are asociat un număr care reprezintă minimul celor două numere asociate fiilor săi.

Pentru a sorta n numere $a_1, \ldots, a_n$ putem proceda după cum urmează :

Determinăm mai întîi $\min(a_1, \ldots, a_n)$ și acest număr îl punem pe primul loc în șirul sortat. Pentru a găsi numărul de pe locul al doilea determinăm minimul din cele $n - 1$ numere rămase ș.a.m.d.

Numărul de comparații necesare pentru a sorta cele n numere cu acest algoritm este egal cu :

$$(n - 1) + (n - 2) + \ldots + 1 = \frac{n(n-1)}{2}.$$

Vom vedea că acest număr poate fi îmbunătățit substanțial prin construcția unui *arbore de sortare* pentru care toate vîrfurile terminale apar pe un singur nivel sau pe două nivele consecutive. Pentru a construi un astfel de arbore, să observăm că dacă arborele este complet și toate vîrfurile terminale apar pe nivelul r, atunci el conține 2^r vîrfuri terminale.

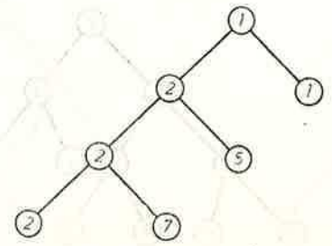


Fig. II.41

Într-adevăr, pe nivelul 1 sînt 2 vîrfuri, pe nivelul 2 sînt 4 vîrfuri şi dacă presupunem că pe nivelul $r-1$ sînt 2^{r-1} vîrfuri, rezultă că pe nivelul r sînt $2 \cdot 2^{r-1} = 2^r$ vîrfuri. Deci dacă numărul de vîrfuri terminale este o putere a lui 2, adică $n = 2^r$, putem construi un arbore binar complet cu toate vîrfurile terminale pe un acelaşi nivel.

În caz contrar, vom nota cu r numărul natural care verifică inegalităţile :

$$2^r < n < 2^{r+1}. \tag{1}$$

Să notăm cu x numărul de vîrfuri care se găsesc pe nivelul $r+1$ într-un arbore binar complet cu toate vîrfurile terminale pe două nivele consecutive şi cu y numărul de vîrfuri care se găsesc pe nivelul r.

Deoarece numărul total de vîrfuri terminale este egal cu n, putem scrie :

$$x + y = n. \tag{2}$$

Dacă construim pentru fiecare vîrf de pe nivelul r cei doi fii care se găsesc pe nivelul $r+1$, obţinem, împreună cu cele x noduri deja existente pe nivelul $r+1$, un total de 2^{r+1} vîrfuri pe nivelul $r+1$. Deci mai obţinem :

$$x + 2y = 2^{r+1}. \tag{3}$$

Ecuaţiile (2) şi (3) formează un sistem care determină complet pe x şi y şi anume : $y = 2^{r+1} - n$ şi $x = 2n - 2^{r+1}$. Deoarece sînt verificate inegalităţile (1) rezultă că $2^{r+1} - n > 0$ şi $2n - 2^{r+1} > 0$, deci vîrfurile arborelui se găsesc pe nivelele r şi $r+1$. De exemplu, dacă $n = 6$ obţinem $2^2 < 6 < 2^3$, deci $r = 2$ şi $y = 2^3 - 6 = 2$ iar $x = 12 - 2^3 = 4$. Arborele binar complet cu 6 vîrfuri terminale pe două nivele consecutive va avea deci 2 vîrfuri pe nivelul 2 şi 4 vîrfuri pe nivelul 3. El este desenat în figura II.42. Să presupunem acum că vrem să sortăm numerele din şirul : 4, 6, 3, 5, 2, 8.

Vom aplica următorul algoritm, numit selecţia arborescentă : reprezentăm aceste 6 numere în vîrfurile terminale ale arborelui din figura II.42. Completăm celelalte vîrfuri interioare ale arborelui, astfel încît în fiecare vîrf interior să se afle scris minimul din cele două numere asociate fiilor săi.

Numărul de comparaţii necesare este egal cu 5, adică numărul minim de comparaţii pentru găsirea minimului din 6 numere.

În vîrful arborelui se află numărul 2, care este minimul din cele 6 numere. Scriem numărul 2 pe prima poziţie a şirului sortat şi scoatem numărul 2 din arborele de sortare, punînd o linioară în vîrful terminal unde a fost scris numărul 2. Pentru a completa arborele şi a găsi minimul din numerele rămase, este suficient să completăm numerele înscrise în vîrfurile lanţului care uneşte rădăcina arborelui cu vîrful terminal unde a fost scris numărul 2.

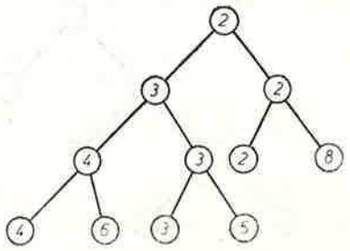


Fig. II.42

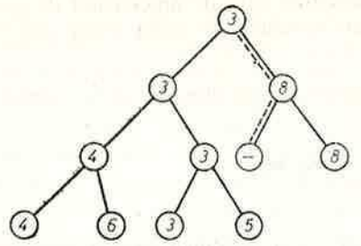


Fig. II.43

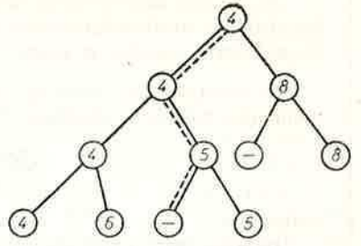


Fig. II.44

Acest lanţ a fost reprezentat cu linie punctată în figura II.43. Numărul 8 nu are cu cine să mai fie comparat pe nivelul 2, astfel încît el urcă pe nivelul 1. Aici el se compară cu 3. Găsim min (3, 8) = 3, pe care îl scriem în rădăcina arborelui. Deci al doilea număr în şirul sortat este 3.

Considerînd lanţul care trece prin vîrfurile cu numărul 3, reprezentat în figura II.44, scoatem numărul 3 din arbore şi mai facem două comparaţii : min (4, 5) = 4 şi min (4, 8) = 4. În rădăcina arborelui apare acum numărul 4, care este al treilea număr din şirul sortat. Se reactualizează lanţul desenat punctat în figura II.45 prin încă două comparaţii : min (5, 6) = 5 şi min (5, 8) = 5, care se ridică în rădăcina arborelui. Al patrulea număr din şirul sortat este 5 şi se obţine arborele din figura II.46 după încă o comparaţie : min (6, 8) = 6. Deci penultimul număr din şirul sortat este 6. În final numărul 8 apare în rădăcina arborelui fără nici o comparaţie şi el este ultimul număr din şirul sortat.

Să evaluăm numărul de comparaţii necesare pentru a sorta n numere, folosind algoritmul de selecţie arborescentă.

Din (1) deducem că $r < \log_2 n < r+1$, deci $r = [\log_2 n]$, adică r este partea întreagă din $\log_2 n$. Dacă $n = 2^r$ obţinem că $r = \log_2 n = [\log_2 n]$, deoarece în acest caz $\log_2 n$ este număr întreg. La primul pas facem $n-1$ comparaţii pentru a completa întregul arbore de sortare şi deci pentru a găsi $\min(a_1, \ldots, a_n)$ care apare în rădăcină. La fiecare din paşii următori facem comparaţii cel mult la nivelele r, $r-1$, ..., 1, pentru a completa vîrfurile care se găsesc pe lanţul care uneşte rădăcina cu vîrful terminal din care a fost suprimat un număr. Pentru exemplul dat, $[\log_2 6] = 2$ şi la fiecare pas ulterior am făcut cel mult $r = 2$ comparaţii.

Deci numărul de comparaţii este majorat de :

$$n - 1 + (n-1)r = (n-1)(r+1) = (n-1)([\log_2 n] + 1).$$

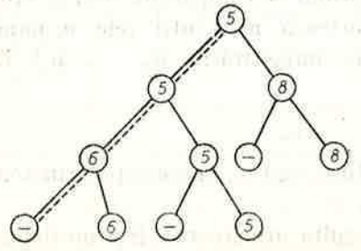


Fig. II.45

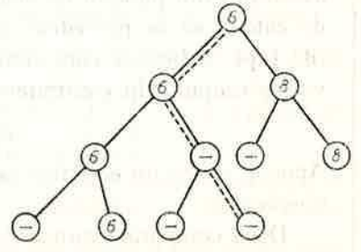


Fig. II.46

Avantajul algoritmului selecției arborescente față de algoritmul de găsire repetată a minimului în ceea ce privește numărul de comparații iese clar în evidență pentru n mare.

De exemplu, dacă avem de sortat 1 001 numere, cu algoritmul de găsire repetată a minimului, trebuie să efectuăm

$$\frac{1\,001(1001-1)}{2} = 500\,500$$

comparații.

Deoarece $2^{10} = 1\,024$, rezultă $r = [\log_2 1\,001] = 9$ și algoritmul selecției arborescente necesită numai $1\,000 \cdot 10 = 10\,000$ comparații, adică de 50 de ori mai puțin.

Încheiem această aplicație menționînd că există metode eficiente de reprezentare a structurii de arbore binar în memoria unui calculator.

II.5.3. Algoritmul de căutare binară

Problema regăsirii informației stocate în memoria unui calculator constă în găsirea unui articol, care conține de obicei mai multe înregistrări memorate secvențial, după anumite valori, numerice sau alfanumerice, ale uneia sau mai multor înregistrări care formează acel articol.

Deci în cazul valorilor numerice, problema găsirii unui articol dintr-o mulțime de articole memorate secvențial în memoria unui calculator, revine în esență la următoarea problemă de căutare:

Dîndu-se n numere într-o ordine fixată, $a_1, a_2, \ldots, a_n$, să se determine care număr are o valoare dată b. Vom presupune că problema are un rezultat unic, deci există cel mult un număr dintre $a_1, \ldots, a_n$ care este egal cu b. Putem rezolva această problemă citind pe rînd fiecare dintre numerele $a_1, a_2, \ldots$ din memoria calculatorului și comparîndu-le cu b. În momentul cînd găsim un număr a_i egal cu b ne oprim sau citim de exemplu și celelalte înregistrări ale articolului care are înregistrarea căutată egală cu a_i. Spunem că în acest caz am avut o căutare cu succes. Dacă însă, după citirea tuturor numerelor $a_1, \ldots, a_n$ am constatat că nici unul dintre ele nu este egal cu b, spunem că a avut loc o căutare fără succes.

În cazul unei căutări cu succes numărul mediu de comparații necesar pentru găsirea valorii b este egal cu $\frac{n}{2}$, iar în cazul unei căutări fără succes numărul de comparații este egal cu n.

Dacă efectuăm multe căutări printre înregistrările $a_1, \ldots, a_n$ este mai avantajos din punctul de vedere al numărului de comparații, deci al vitezei de calcul, să se procedeze astfel: se sortează mai întîi cele n numere (de fapt articolele care conțin respectiv înregistrările $a_1, \ldots, a_n$). Deci vom presupune în continuare că:

$$a_1 < a_2 < \ldots < a_n.$$

Apoi se aplică un algoritm rapid de căutare, care se bazează pe următoarea observație:

Dacă comparăm numărul a_i cu b, rezultă următoarele trei posibilități:

a) $a_i = b$ și căutarea este încheiată;

b) $a_i < b$ și deci vom continua căutarea printre numerele $a_{i+1} < a_{i+2} < \ldots < a_n$, care sînt mai mari ca a_i. Dacă $i = n$, deci nu mai există numere mai mari ca a_i, ne oprim, deoarece căutarea nu a avut succes. Numărul b nu se găsește printre numerele $a_1, \ldots, a_n$;

c) $a_i > b$ și deci vom continua căutarea printre numerele $a_1 < a_2 < \ldots < a_{i-1}$, care sînt mai mici ca a_i. Dacă $i = 1$, deci nu mai există numere mai mici ca a_i ne oprim, deoarece căutarea nu a avut succes. Rezultă că b nu se găsește printre numerele date. Algoritmul de căutare care va fi prezentat în continuare alege mereu numărul a_i la mijlocul șirului de numere în care căutăm numărul b. De exemplu, dacă n este impar, numărul de indice $\frac{n+1}{2}$ se găsește la mijlocul șirului $a_1, \ldots, a_n$. Dacă însă n este par, există două numere care se găsesc la mijlocul șirului și anume numerele de indici $\frac{n}{2}$ și $\frac{n}{2}+1$. Deoarece n este par obținem $\frac{n}{2} = \left[\frac{n+1}{2}\right]$. Algoritmul de căutare binară utilizează mereu valoarea $\left[\frac{p+q}{2}\right]$ pentru indicele numărului de la mijlocul șirului:

$$a_p < a_{p+1} < \ldots < a_q.$$

Algoritmul de căutare binară.

Fiind date numerele în ordine crescătoare $a_1 < a_2 < \ldots < a_n$, se caută în acest șir numărul b.

1) Se stabilește $p \leftarrow 1$, $q \leftarrow n$.

2) $q < p$? Da: Stop. Algoritmul se termină fără succes. În caz contrar se stabilește $i \leftarrow \left[\frac{p+q}{2}\right]$.

3) Se compară b cu a_i. Dacă $b < a_i$, se merge la pasul 4); dacă $b > a_i$, se merge la 5), iar dacă $b = a_i$, algoritmul se termină cu succes.

4) Se atribuie $q \leftarrow i - 1$ și se merge la 2.

5) Se atribuie $p \leftarrow i + 1$ și se merge la 2.

La pașii 4 și 5 se redefinesc marginile subșirului care trebuie cercetat în continuare, în funcție de rezultatul comparației de la 3. Condiția de oprire în cazul căutării fără succes este $q < p$, așa cum vom verifica pe un exemplu.

Să aplicăm algoritmul de căutare binară pentru găsirea numărului 25 din șirul sortat:

3, 5, 12, 14, 15, 18, 24, 25, 42.

Deci $n = 9$, $a_1 = 3$, $a_2 = 5$, $\ldots$, $a_8 = 25$, $a_9 = 42$. Inițial $p = 1$, $q = 9$, $i = 5$. La pasul 3 găsim $25 > a_5 = 15$, deci la pasul 5 redefinim $p = 6$. Se merge la 2 și se calculează $i = \left[\frac{15}{2}\right] = 7$. La pasul 3 găsim $25 > a_7 = 24$, deci trecem la pasul 5 unde redefinim $p = 8$.

Ne întoarcem la pasul 2 unde calculăm noua valoare a lui $i = \left[\frac{17}{2}\right] = 8$, iar la pasul 3 găsim $25 = a_8$. Algoritmul s-a terminat cu succes după trei comparații.

Să presupunem acum că vrem să căutăm (fără succes) numărul 13. Inițial $p = 1$, $q = 9$, $i = 5$. La pasul 3 găsim $13 < a_5 = 15$, deci la pasul 4 redefinim $q = 4$. Mergem la 2 și calculăm $i = \left[\frac{5}{2}\right] = 2$. La pasul 3 găsim $13 > 5 = a_2$, deci trecem la pasul 5 unde rede-

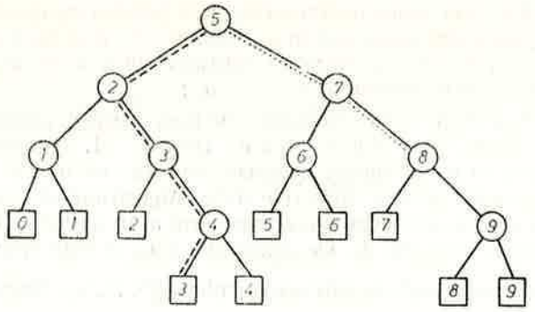


Fig. II.47

finim $p = 3$. Ne întoarcem la 2 unde calculăm $i = \left[\frac{7}{2}\right] = 3$. La 3 găsim $13 > a_3 = 12$, deci mergem la pasul 5 unde obținem $p = 4$. Ne întoarcem la 2 cu valorile $p = q = 4$, deci $i = 4$, iar la 3 găsim $13 < a_4 = 14$. Mergem la 4 unde dăm valoarea $q = 3$.

Ne întoarcem la 2 unde găsim $q < p$, deci algoritmul s-a terminat fără succes după efectuarea a 4 comparații.

În cazul unei căutări fără succes a numărului b, la pasul 2 condiția $q < p$ este verificată cînd $p = r$, $q = r - 1$ și există inegalitățile:

$$a_{r-1} < b < a_r.$$

Într-adevăr, algoritmul localizează numărul b între numerele de indici $r - 1$ și r, deci la un moment dat $p = r - 1$ și $q = r$. Deoarece $b > a_{r-1}$, i se dă lui p valoarea r. Găsim $b < a_r$, deci q ia valoarea $r - 1$ și ne oprim deoarece $q < p$. Algoritmul de căutare binară poate fi privit ca un arbore de decizie binar, ca cel din figura II.47 pentru cazul $n = 9$.

În acest caz prima comparație efectuată este aceea a lui b cu a_5, care este reprezentată prin nodul rădăcină al arborelui binar din figura II.47. Dacă găsim $b = a_5$ ne oprim. În caz contrar, dacă $b < a_5$ vom compara pe b cu a_2, comparație care este reprezentată prin rădăcina subarborelui stîng al rădăcinii 5. Dacă însă $b > a_5$, vom compara pe b cu a_7, comparație care este reprezentată prin rădăcina subarborelui drept al rădăcinii 5. În fiecare vîrf al arborelui care se obține în acest mod este scris indicele i al numărului a_i cu care se compară b în acel moment. Dacă $b < a_i$ se urmează ramura din stînga, efectuînd o comparație cu numărul al cărui indice este scris în rădăcina subarborelui stîng al nodului i. Dacă $b > a_i$ se urmează ramură din dreapta, efectuînd o comparație cu numărul al cărui indice este scris în rădăcina subarborelui drept al nodului i. O căutare fără succes va conduce la unul din nodurile terminale, reprezentate printr-un pătrat, numerotate de la 0 la 9 în sensul de la stînga spre dreapta.

De exemplu, se ajunge la nodul 0 dacă $b < a_1$, la nodul 1 dacă $a_1 < b < a_2$, ..., la nodul 9 dacă $b > a_9$.

Căutarea cu succes a numărului 25 a fost reprezentată în figura II.47 prin linie punctată și corespunde parcurgerii lanțului care unește rădăcina arborelui cu nodul neterminal cu numărul 8. Căutarea fără succes a numărului $b = 13$ a fost reprezentată prin linie întreruptă și corespunde parcurgerii lanțului care unește rădăcina cu nodul terminal cu numărul 3, deoarece $a_3 < b < a_4$. Se observă că în fiecare caz numărul de comparații este egal

cu numărul de noduri neterminale care se găsesc pe lanțul care unește rădăcina cu vîrful care reprezintă sfîrșitul procesului de căutare. Astfel, în cazul căutării numărului $25 = a_8$ s-au efectuat trei comparații, iar în cazul căutării fără succes a numărului $a_3 < 13 < a_4$ au fost efectuate patru comparații: cu a_5, cu a_2, cu a_3 și cu a_4.

Reprezentarea strategiei de căutare în cazul algoritmului de căutare binară printr-un *arbore de decizie binar* ne va ajuta la evaluarea performanțelor acestui algoritm. Folosim termenul de arbore de decizie binar deoarece decizia de continuare a procesului de căutare, adică fie oprirea căutării, fie căutarea pe ramura din stînga, fie căutarea pe ramura din dreapta a unui nod, depind de rezultatul comparației numărului căutat cu numărul din șir indicat de nodul în care ne aflăm. Evaluarea eficienței acestui algoritm este dată de următoarea teoremă, care poate fi demonstrată prin inducție după k.

Teoremă. *Dacă efectuăm o căutare într-un șir ordonat de n numere cu algoritmul de căutare binară și dacă n verifică inegalitățile:*

$$2^{k-1} \leqslant n < 2^k, \tag{1}$$

atunci:

a) *O căutare cu succes necesită cel mult k comparații;*

b) *O căutare fără succes necesită $k - 1$ sau k comparații.*

O ilustrare a acestei proprietăți pentru cazul $n = 4$ este dată în figura II.48. Observăm că numărul maxim de comparații în cazul unei căutări cu succes este egal cu numărul de nivele pe care se găsesc nodurile neterminale, adică cu 3. Se vede din figura II.48 că acest număr maxim de comparații se atinge numai pentru găsirea ultimului număr a_4.

În cazul unei căutări fără succes numărul de comparații este egal cu 2 sau cu 3, deoarece nodurile terminale se găsesc pe nivelele 2 și 3. Din figura II.48 se observă că în acest caz numărul de comparații egal cu 3 se atinge numai pentru vîrfurile terminale 3 și 4, deci în cazurile cînd $a_3 < b < a_4$ și respectiv $b > a_4$.

Din (1) deducem $k - 1 \leqslant \log_2 n < k$, deci $k - 1 = [\log_2 n]$ de unde $k = [\log_2 n] + 1$.

De exemplu, dacă vrem să căutăm un număr printre 1 000 de înregistrări citind înregistrare cu înregistrare și făcînd comparațiile respective, trebuie să facem în medie 500 de comparații în cazul unei căutări cu succes și 1 000 de comparații în cazul unei căutări fără succes.

Dacă însă înregistrările sînt sortate, în cazul folosirii algoritmului de căutare binară vom evalua numărul de comparații aplicînd teorema anterioară.

Deoarece $2^{10} = 1\,024$, obținem $k = [\log_2 1\,000] + 1 = 10$. Deci orice căutare cu succes necesită între una și zece comparații, iar orice căutare fără succes necesită 9 sau 10 comparații.

S-a obținut o reducere importantă a numărului de comparații, deci implicit și a timpului de căutare, în raport cu cazul înregistrărilor nesortate.

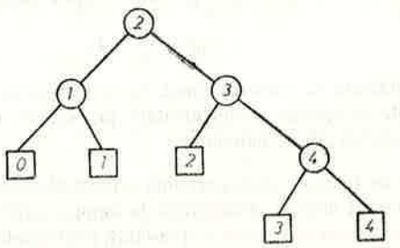


Fig. II.48

Probleme

II.5.1. Să se scrie notația fără paranteze pentru fiecare din următoarele expresii aritmetice:

a. $(x + y)(y + z)(z + x)$;

b. $3x^3y - 5x^2y^2 + 2xy^3$;

c. $\frac{x+y}{y+z} + \frac{y+z}{z+x} + \frac{z+x}{x+y}$;

d. $a^3 + b^3 + c^3 - 3abc$;

e. $(2x^{3y+z} - a)(14t - 3)$;

f. $a^{b^c} + \dfrac{x}{2 + \frac{y}{z}}$.

II.5.2. Să se arate că orice arbore binar complet cu n vîrfuri terminale are în total $2n - 1$ vîrfuri. Să se deducă de aici o relație simplă între numărul de operatori și numărul de operanzi dintr-o expresie aritmetică.

II.5.3. Să se deducă scrierea obișnuită a următoarelor expresii aritmetice, date în notația fără paranteze, știind că toate constantele numerice utilizate au o singură cifră:

a. $- - * * * 5\ x\ y\ z * 4 * \uparrow x\ 5\ y * 3\ z$

b. $/\ x + y\ /\ \uparrow\ x\ 2\ z$

c. $\uparrow\ x + a + * 2\ b * 3\ \uparrow\ c\ 2$

d. $* + * + x * 3\ a - y\ 1\ 2 + \uparrow\ x\ 2\ 5$.

II.5.4. Să se construiască un arbore de sortare pentru cazul $n = 10$.

II.5.5. Să se dea o definiție matematică pentru relația de ordine lexicografică (ordinea scrierii într-un dicționar) a cuvintelor formate cu litere dintr-un alfabet cu 24 de litere.

II.5.6. Pe o bandă magnetică sînt înscrise un milion de numere. Indicați un algoritm de determinare a numărului de numere distincte de pe bandă.

II.5.7 Se consideră un fișier cu 4 001 cuvinte binare distincte de 30 biți: $a_1, a_2, \ldots, a_{4001}$. Două cuvinte binare $x = x_1 \ldots x_{30}$ și $y = y_1 \ldots y_{30}$ se numesc complementare dacă

$$x_i + y_i = 1 \text{ pentru } i = 1, \ldots, 30.$$

Cu alte cuvinte, putem spune că x și y sînt complementare dacă și numai dacă

$$x + y = 11 \ldots 1_2,$$

ele fiind considerate ca numere binare. Să se definească un algoritm pentru a găsi toate perechile de cuvinte complementare $\{a_i, a_j\}$, cu un număr cît mai redus de operații (de comparație și adunare).

II.5.8. La un turneu de tenis de cîmp participă n jucători. Cum trebuie organizat turneul, care conține numai meciuri eliminatorii de simplu, astfel încît dacă pentru determinarea campionului sînt necesare $n - 1$ meciuri, pentru determinarea celui de al II-lea jucător al turneului să mai fie necesară disputarea a numai $[\log_2 n]$ meciuri?

II.5.9. Se consideră următorul algoritm de sortare prin interschimbare a șirului de numere $a_1, a_2, \ldots, a_n$:

1. $i \leftarrow n$;

2. $t \leftarrow 0$. Efectuează pasul 3 pentru $j = 1, 2, \ldots, i - 1$ și apoi execută pasul 4.

3. Dacă $a_j > a_{j+1}$, interschimbă numerele a_j și a_{j+1} în șir (a_j este trecut pe poziția $j + 1$, deci el va fi notat cu a_{j+1} și a_{j+1} trece pe poziția j, fiind notat în continuare cu a_j). Se stabilește $t \leftarrow j$.

4. Dacă $t = 0$ ne oprim. Șirul obținut este sortat. În caz contrar $i \leftarrow t$ și se revine la pasul 2.

Să se justifice acest algoritm, considerînd mai multe exemple de aplicare și să se arate că are un număr finit de pași.

II.5.10. Se consideră două șiruri sortate de numere:

$$a_1 \leqslant a_2 \leqslant \ldots \leqslant a_m \text{ și } b_1 \leqslant b_2 \leqslant \ldots \leqslant b_n.$$

Să se propună un algoritm de *interclasare* a celor două șiruri de numere într-un singur șir sortat $c_1 \leqslant c_2 \leqslant \ldots \leqslant c_{m+n}$, format din elementele celor două șiruri, care să folosească cel mult $m + n - 1$ comparații.

II.5.11. Ce algoritm de căutare corespunde arborelui binar complet din figura II.49 în cazul unui șir sortat format din 4 numere?

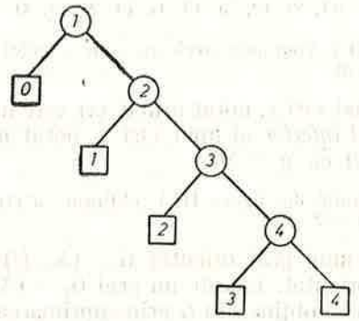


Fig. II.49

II.5.12. Să se calculeze numărul mediu de comparații în cazul unei căutări cu succes și în cazul unei căutări fără succes pentru arborii de căutare din figurile II.48 și II.49.

II.5.13. O bandă magnetică conține n înregistrări.

Se presupune că frecvența acceselor la cele n înregistrări este diferită, adică unele sînt citite mai des, altele mai rar. Se știe că timpul de citire a unei înregistrări este cu atît mai mare cu cît înregistrarea respectivă se găsește mai departe de începutul benzii, deoarece crește timpul necesar derulării benzii magnetice.

Să se indice o strategie simplă de auto-organizare a datelor pe banda magnetică, astfel încît timpurile medii de acces la înregistrări să fie cît mai mici.

III. GRAFURI ORIENTATE

III.1. Noţiuni de bază

Un *graf orientat* G este format dintr-o pereche ordonată de mulţimi $G = (X, U)$. Ca şi în cazul grafurilor neorientate, X este *mulţimea vîrfurilor* sau *nodurilor* grafului. Mulţimea U este formată din perechi ordonate de elemente distincte din X, numite *arce*. Orice arc $u \in U$ va fi notat prin $u = (x, y)$ cu $x, y \in X$ şi $x \neq y$.

Spunem că vîrful x este *extremitatea iniţială* a arcului u, iar vîrful y este *extremitatea finală* a arcului u. Spre deosebire de cazul grafurilor neorientate, notaţiile (x, y) şi (y, x) vor desemna două arce diferite.

Dacă graful G conţine arcul (x, y) vom spune că vîrfurile x şi y sînt *adiacente* în G şi amîndouă sînt *incidente* cu arcul (x, y). Deci un graf orientat G poate fi imaginat ca o mulţime de vîrfuri, dintre care unele sînt unite două cîte două prin arce. Un graf orientat poate fi desenat în plan reprezentînd vîrfurile sale prin puncte şi arcele prin săgeţi care sînt orientate de la extremitatea iniţială către extremitatea finală a fiecărui arc.

Graful orientat $G = (X, U)$ unde:
$X = \{1, 2, \ldots, 8\}$ cu $U = \{(1, 2), (2, 3), (3, 1), (3, 2), (2, 4), (4, 5), (3, 5), (6, 8), (8, 7), (7, 8), (7, 6)\}$
se reprezintă ca în figura III.1. Vom nota arcele aşa cum se indică în figură, adică $u_1 = (1, 2)$, $u_2 = (3, 1)$, $\ldots$, $u_{11} = (6, 8)$.

Gradul exterior al unui vîrf x, notat prin $d^+(x)$, este numărul arcelor de forma (x, y) cu $y \in X$. *Gradul interior* al unui vîrf x, notat prin $d^-(x)$, este numărul arcelor de forma (y, x) cu $y \in X$.

De exemplu, pentru graful din figura III.1, obţinem: $d^+(1) = 1$, $d^-(1) = 1$, $d^+(2) = 2$, $d^-(2) = 2$, $d^+(5) = 0$, $d^-(5) = 2$ etc.

Un *graf parţial* al unui graf orientat $G = (X, U)$ se defineşte în acelaşi mod ca şi în cazul neorientat. El este un graf $G_1 = (X, V)$ unde $V \subset U$, deci este graful G însuşi sau se obţine din G prin suprimarea anumitor arce.

Şi definiţia unui subgraf al unui graf orientat $G = (X, U)$ este asemănătoare cu cazul neorientat. Prin definiţie, un *subgraf* al lui G este un graf $H = (Y, V)$

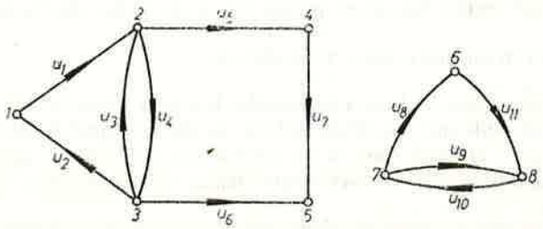


Fig. III.1

unde $Y \subset X$ iar arcele din V sînt toate arcele din U care au ambele extremităţi în mulţimea de vîrfuri Y.

Deci un subgraf H al unui graf orientat G este graful G însuşi sau se obţine din G prin suprimarea anumitor vîrfuri şi a tuturor arcelor incidente cu acestea.

Vom spune că subgraful H este *indus* sau *generat* de mulţimea de vîrfuri Y.

Astfel, subgraful grafului G din figura III.1, indus de mulţimea de vîrfuri $Y_1 = \{1, 2, 4, 5\}$ are ca mulţime de arce mulţimea $V_1 = \{(1, 2), (2, 4), (4, 5)\}$, iar subgraful indus de mulţimea de vîrfuri $Y_2 = \{6, 7, 8\}$ are mulţimea arcelor $V_2 = \{(7, 6), (6, 8), (7, 8), (8, 7)\}$.

Un graf orientat este *complet* dacă oricare două vîrfuri sînt adiacente. De exemplu, subgraful grafului din figura III.1, indus de mulţimea de vîrfuri Y_2, este complet.

În timp ce în cazul neorientat un graf complet cu n vîrfuri este unic determinat, în cazul orientat există mai multe grafuri complete cu un număr dat de vîrfuri. Ele se deosebesc fie prin orientarea arcelor, fie prin faptul că între două vîrfuri oarecare există un arc sau două arce de sensuri contrare.

Un *lanţ* al unui graf orientat se defineşte ca un şir de arce:

$$L = [u_1, u_2, \ldots, u_p]$$

cu proprietatea că oricare două arce vecine u_i şi u_{i+1} au o extremitate comună pentru orice $i = 1, \ldots, p - 1$.

Extremitatea x_0 a lui u_1 care nu este comună cu u_2 şi extremitatea x_r a lui u_p care nu este comună cu u_{p-1} se numesc *extremităţile lanţului* L.

Dacă toate arcele lanţului L au o aceeaşi orientare, care este dată de sensul deplasării de la x_0 către x_r, lanţul se numeşte *drum*.

Deci un drum într-un graf orientat $G = (X, U)$ este un şir de vîrfuri notat:

$$D = (x_0, x_1, \ldots, x_r)$$

cu proprietatea că $(x_0, x_1), (x_1, x_2), \ldots, (x_{r-1}, x_r) \in U$, deci sînt arce ale grafului.

Vîrfurile x_0 şi x_r se numesc extremităţile drumului D. Dacă vîrfurile $x_0, x_1, \ldots, x_r$ sînt distincte două cîte două, drumul D se numeşte *elementar*. Din aceste definiţii rezultă că orice drum este şi lanţ, dacă îl privim ca un şir de arce.

Pentru graful din figura III.1 următoarele şiruri de arce sînt lanţuri:
$L_1 = [u_1, u_3, u_6, u_7]$, $L_2 = [u_4, u_5]$, $L_3 = [u_8, u_{10}]$, $L_4 = [u_5, u_4, u_6, u_3, u_1]$, $L_5 = [u_8, u_{11}]$
Lanţurile L_2 şi L_5 sînt chiar drumuri şi ele pot fi scrise ca un şir de vîrfuri în modul următor: (2, 3, 2), respectiv (7, 6, 8).
L_1 are extremităţile 1 şi 4, L_6 are extremităţile 6 şi 8, L_4 are extremităţile 2 şi 1, iar pentru L_2 cele două extremităţi coincid cu vîrful 2. Pentru acelaşi graf următoarele şiruri de vîrfuri sînt drumuri:
$D_1 = (1, 2, 4, 5)$, $D_2 = (7, 6, 8, 7)$, $D_3 = (3, 1, 2, 3, 5)$, $D_4 = (3, 2, 4, 5)$, $D_5 = (2, 3, 2, 3, 1, 2)$.
Drumurile D_1 şi D_4 sînt drumuri elementare, deoarece nu trec de două ori printr-un acelaşi vîrf.

Un drum $D = (x_0, \ldots, x_r)$ poate fi interpretat ca traseul unei deplasări pe arcele grafului în ordinea $(x_0, x_1), (x_1, x_2), \ldots, (x_{r-1}, x_r)$.

De aceea drumul D de extremităţi x_0 şi x_r se mai spune că este un drum de la x_0 la x_r. Dacă $x_0 = x_r$ şi toate arcele $(x_0, x_1), (x_1, x_2), \ldots, (x_{r-1}, x_r)$ sînt distincte două cîte două, drumul D se numeşte *circuit*.

Dacă toate vîrfurile circuitului, cu excepţia primului şi a ultimului vîrf, sînt distincte două cîte două, circuitul se numeşte *elementar*.

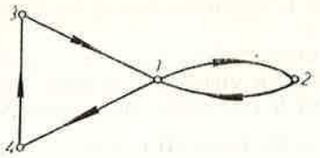


Fig. III.2

De exemplu drumul D_1 nu este circuit, deoarece foloseşte de două ori acelaşi arc u_4. Drumul D_2 este un circuit elementar. Considerînd graful orientat din figura III.2, un circuit care nu este elementar este următorul : $C = (1, 2, 1, 4, 3, 1)$, care trece de două ori prin vîrful 1.

Ca şi în cazul ciclurilor din grafurile neorientate, circuitele pot fi scrise în mai multe moduri, alegînd ca prim vîrf un vîrf arbitrar prin care trece circuitul.

De exemplu, circuitul C din figura III.2 mai poate fi scris :
$C = (2, 1, 4, 3, 1, 2)$ sau $C = (4, 3, 1, 2, 1, 4)$ sau
$C = (1, 4, 3, 1, 2, 1)$ sau $C = (3, 1, 2, 1, 4, 3)$.

Noţiunile de *conexitate* şi de *componentă conexă* a unui graf orientat sînt similare cu cele de la grafurile neorientate, utilizînd noţiunea de lanţ din cazul grafurilor orientate.

Astfel, un graf orientat G se numeşte *conex* dacă pentru oricare două vîrfuri distincte x şi y există un lanţ de extremităţi x şi y în G. O *componentă conexă* C a unui graf orientat G se defineşte ca fiind un subgraf conex maximal al lui G, deci nu există nici un lanţ care să unească un vîrf din C cu un vîrf care nu aparţine lui C.

Graful din figura III.1 nu este conex şi are 2 componente conexe : C_1 indusă de mulţimea de vîrfuri $\{1, 2, 3, 4, 5\}$ şi C_2 indusă de mulţimea de vîrfuri $\{6, 7, 8\}$.

Graful din figura III.2 este însă conex, deci are o singură componentă conexă.

Probleme

III.1.1. Să se găsească drumurile elementare de la vîrful 1 la vîrful 8 pentru graful din figura III.3. Care sînt circuitele elementare ale acestui graf ?

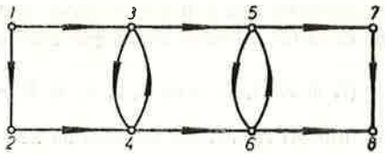


Fig. III.3

III.1.2. Să se arate că dacă graful orientat G cu mulţimea de vîrfuri X are m arce, au loc egalităţile :

$$\sum_{x \in X} d^-(x) = \sum_{x \in X} d^+(x) = m.$$

III.1.3. Să se calculeze numărul grafurilor orientate cu n vîrfuri date : $x_1, \ldots, x_n$. Cîte grafuri orientate şi complete cu n vîrfuri date există ?

III.1.4. Pe mulţimea X a vîrfurilor unui graf orientat $G = (X, U)$ se introduce următoarea relaţie binară : Spunem că x este în relaţie cu y şi scriem $x \sim y$ dacă $x = y$ sau dacă există un drum de la x la y şi un drum de la y la x.
Să se arate că această relaţie binară este o relaţie de echivalenţă. Clasele acestei echivalenţe se numesc *componentele tare conexe* ale grafului. Să se determine componentele tare conexe ale grafului din figura III.1.

III.1.5. Fie $D = (x, \ldots, y)$ un drum de la x la y $(x \neq y)$ în graful G. Să se arate că există un drum elementar de la x la y în G.

III.1.6. Un graf orientat cum este cel din figura III.4, care are proprietatea că între oricare două vîrfuri distincte x şi y există un arc şi numai unul se numeşte *graf-turneu*.

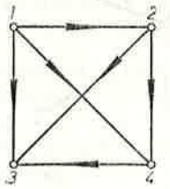


Fig. III.4

Un graf-turneu poate reprezenta rezultatele meciurilor directe între participanţii la o competiţie sportivă în care nu există meciuri nule şi nici meciuri eliminatorii. Să se arate că orice graf-turneu conţine un drum elementar care trece prin toate vîrfurile grafului.

III.1.7. Pentru un graf-turneu cu n vîrfuri $x_1, \ldots, x_n$ se notează $r_i = d^-(x_i)$ şi $s_i = d^+(x_i)$ pentru $i = 1, \ldots, n$. Să se arate că :

1) $r_i + s_i = n - 1$

2) $\sum_{i=1}^{n} r_i = \sum_{i=1}^{n} s_i = C_n^2$

3) $\sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} s_i^2$.

III.1.8. Să se arate că pentru orice graf-turneu G există un vîrf x, astfel încît toate vîrfurile y diferite de x ale grafului pot fi atinse plecînd din x pe drumuri care au un arc sau două arce.

III.2. Metoda drumului critic

III.2.1. Drumul critic într-un graf de activităţi

Una dintre aplicaţiile cele mai răspîndite ale teoriei grafurilor în probleme de economie o constituie *metoda drumului critic*.

O lucrare complexă a cărei realizare comportă mai multe activităţi, se poate reprezenta printr-un graf orientat care dă o imagine grafică a eşalonării în timp şi a intercondiţionării tuturor activităţilor elementare care alcătuiesc întreaga lucrare.

Pentru un astfel de graf, arcele reprezintă *etapele* sau *activităţile elementare* ale lucrării. Fiecare arc are o anumită lungime, care reprezintă timpul de desfăşurare a activităţii asociate acelui arc.

Momentul începerii activității este reprezentat de extremitatea inițială, iar momentul terminării activității este reprezentat de extremitatea finală a arcului respectiv.

Astfel pentru graful de activități din figura III.5 începerea întregii lucrări este reprezentată prin vîrful A, iar terminarea ei prin vîrful H.

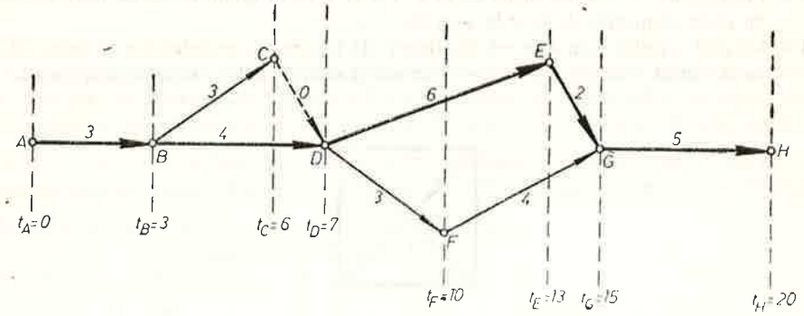


Fig. III.5

În această lucrare anumite activități se desfășoară în același timp, de exemplu activitățile reprezentate de arcele (B, C) și (B, D).

La fel activitățile reprezentate de arcele (D, E) și (E, G) se desfășoară în același timp cu activitățile reprezentate de arcele (D, F) și (F, G).

Nodurile în acest graf de activități reprezintă *evenimente*, care pot fi interpretate ca indicînd realizarea unor obiective parțiale ale lucrării.

Astfel, vîrful A reprezintă începerea întregii lucrări, vîrful B reprezintă evenimentul care constă în terminarea activității reprezentate de arcul (A, B) și începerea activităților reprezentate de arcele (B, C) și (B, D), nodul D reprezintă terminarea activităților reprezentate de arcele (C, D) și (B, D) și începerea activităților reprezentate de arcele (D, E) și (D, F) etc.

Activitatea reprezentată de arcul (C, D), desenat cu linie întreruptă, este o activitate fictivă, care are asociat un timp egal cu zero și reprezintă pur și simplu o relație de anterioritate între terminarea operației (B, C) și începerea operațiilor reprezentate de arcele (D, E) și (D, F).

La definirea grafului care reprezintă succesiunea în timp a activităților unei lucrări complexe, pentru orice vîrf x trebuie să fie îndeplinită următoarea regulă :

Toate arcele care pleacă din x reprezintă operații care nu pot începe decît după terminarea tuturor operațiilor reprezentate de arcele care sosesc în vîrful x.

Activitățile fictive se introduc pentru a nu avea mai multe arce paralele și de același sens între două vîrfuri ale grafului.

Un graf de activități are o proprietate importantă și anume aceea că nu conține circuite, deoarece în caz contrar, conform regulii introduse, o aceeași operație ar trebui să înceapă după terminarea ei însăși, ceea ce este absurd. Într-adevăr, dacă într-un graf de activități, ar exista un circuit :

$$C = (x, y_1, y_2, \ldots, y_p, x)$$

aceasta ar însemna că operația (y_p, x) începe după terminarea operației (y_{p-1}, y_p). Însă operația (y_{p-1}, y_p) poate să înceapă numai după terminarea operației (y_{p-2}, y_{p-1}) ș.a.m.d. Găsim că operația (y_p, x) poate începe numai după terminarea operației (x, y_1), ceea ce contrazice convenția de alcătuire a unui graf de activități.

Duratele operațiilor sau *timpii operatori*, care sînt numere reale nenegative, sînt asociați arcelor grafului, care reprezintă operațiile lucrării.

Astfel arcului (A, B) i se asociază timpul 3, arcului (B, C) timpul 3, arcului (B, D) timpul 4 etc., unitatea de măsură a timpului fiind ziua, săptămîna, luna etc.

Acești operatori timpi vor fi interpretați drept lungimi ale arcelor respective. Lungimea unui drum într-un graf de activități se definește ca fiind egală cu suma lungimilor arcelor care compun acel drum, deci este egală cu timpul necesar pentru executarea tuturor operațiilor cuprinse în drumul respectiv.

Stabilirea grafului de activități necesită pentru fiecare operație cunoașterea duratei sale cît și a operațiilor care o preced nemijlocit, deci toate relațiile de ordine temporală privitoare la aceasta, sau antecedentele obligatorii.

Pentru graful din figura III.5 succesiunea în timp a operațiilor a fost marcată prin linii verticale întrerupte, care ne indică evoluția realizării în timp a întregii lucrări. Astfel evenimentul A are data $t_A = 0$, evenimentul B are data $t_B = 3$, evenimentul C are data $t_C = 6$ și evenimentul D are data $t_D = 7$, deoarece pentru producerea evenimentului D trebuie ca ambele activități (B, C) și (B, D) să fie terminate. Deci $t_D = \max(t_B + 4, t_B + 3) = \max(7, 6) = 7$. În continuare găsim $t_F = t_D + 3 = 10$, $t_E = t_D + 6 = 13$, $t_G = \max(t_E + 2, t_F + 4) = \max(15, 14) = 15$ și $t_H = t_G + 5 = 20$.

Rezultă că timpul total de execuție a întregii lucrări este de 20 unități de timp.

Considerînd că evenimentul care constă în începerea întregii lucrări are data zero, timpul total de execuție a lucrării este data realizării ansamblului de lucrări.

Această dată este egală cu suma timpilor operatori considerați pe drumul cel mai nefavorabil de la A la H, adică acel drum care dă între aceste două vîrfuri o sumă maximă de timpi operatori.

Acest drum, care poate să nu fie unic, este chiar drumul de lungime maximă de la A la H, numit și *drum critic*. Drumul critic pentru graful din figura III.5 a fost desenat cu linii îngroșate și el este (A, B, D, E, G, H). Drumul critic reunește activități de a căror realizare la timpul prevăzut depinde realizarea la timp a întregii lucrări.

Orice mărire a timpului asociat arcelor drumului critic conduce la mărirea timpului de realizare a întregii lucrări.

De aceea în cursul execuției lucrării activitățile situate pe drumul critic și numite activități critice, trebuie supravegheate cu deosebită grijă, pentru a nu depăși timpul planificat de realizare a lucrării.

Calculul datei de realizare a evenimentului final H revine deci la căutarea în graf a drumului celui mai lung sau a drumului critic de la A la H. Acest drum are o lungime de 20 unități de timp și am văzut că el reprezintă timpul de realizare a lucrării pornind de la data zero, care este atribuită evenimentului inițial A.

Deci dacă lucrarea se desfășoară fără incidente, durata sa va fi egală cu 20 unități de timp, de exemplu 20 de luni.

Operațiile critice sînt reprezentate de arcele (A, B), (B, D), (D, E), (E, G) și (G, H) ale drumului critic.

III.2.2. **Evenimente critice și intervale de fluctuație**

Să considerăm un graf de activități G, ale cărui vîrfuri reprezintă evenimentele $E_1, E_2, \ldots, E_n$, vîrfuri care vor fi notate la fel cu evenimentele pe care le reprezintă. Presupunem că E_1 reprezintă începutul lucrării și E_n reprezintă

terminarea ei. Pentru a calcula datele de realizare ale diferitelor evenimente E_i trebuie să determinăm cele mai lungi drumuri în graful de activități de la E_1 la celelalte vîrfuri E_i, pentru a fi siguri că toate operațiile anterioare evenimentului E_i au fost terminate.

Lungimea maximă a drumurilor de forma $D = (E_1, \ldots, E_i)$ din graful de activități se numește *data așteptată* a evenimentului E_i și se notează cu t_i.

Astfel pentru graful de activități din figura III.5 vom alege $t_A = 0$. Data așteptată a evenimentului B va fi $t_B = 3$, deoarece există un singur drum de la A la B și anume arcul (A, B).

De la A la D există două drumuri, însă drumul (A, B, D) are lungimea maximă egală cu 7, deci $t_D = 7$. De la A la C există un singur drum de lungime 6, deci $t_C = 6$. La fel obținem $t_F = 10$ și $t_E = 13$. De la A la G există patru drumuri în graf și anume (A, B, C, D, E, G); (A, B, C, D, F, G); (A, B, D, F, G) și (A, B, D, E, G), însă numai ultimul are o lungime maximă, egală cu 15. Deci $t_G = 15$. Am văzut că lungimea maximă a drumurilor de la A la H este egală cu 20, deci $t_H = 20$.

Este util să cunoaștem pentru fiecare eveniment E_i *data sa limită de realizare*, dată a cărei depășire va determina întîrzierea întregii lucrări. Timpul necesar pentru realizarea operațiilor situate între E_i și evenimentul final E_n se obține căutînd în graf drumul cel mai lung de la E_i la E_n, deoarece trebuie să fim siguri că operațiile care urmează după evenimentul E_i pot fi toate realizate, ținînd seama de duratele lor.

Deci data limită a evenimentului E_i, care se notează t_i^*, se obține scăzînd din data t_n a evenimentului final E_n lungimea maximă a drumurilor $D = (E_i, \ldots, E_n)$.

Pentru graful de activități din figura III.5 obținem $t_H^* = t_H = 20$. Există un singur drum de la G la H și anume arcul (G, H) de lungime 5, deci $t_G^* = 20 - 5 = 15$. Deoarece drumurile (E, G, H), respectiv (F, G, H) au o lungime egală cu 7, respectiv 9, obținem $t_E^* = 20 - 7 = 13$ și $t_F^* = 20 - 9 = 11$.

Cel mai lung drum de la D la H este (D, E, G, H) de lungime egală cu 13, deci $t_D^* = 20 - 13 = 7$.

În mod analog $t_C^* = 7$, $t_B^* = 3$ și $t_A^* = 0$. Pentru evenimentele critice A, B, D, E, G, H, se observă că aceste două date coincid:

$$t_A = t_A^* = 0,\ t_B = t_B^* = 3,\ t_D = t_D^* = 7,\ t_E = t_E^* = 13,\ t_G = t_G^* = 15 \text{ și } t_H = t_H^* = 20.$$

Această proprietate are loc pentru toate evenimentele critice dintr-un graf de activități. Pentru a o justifica, să presupunem că evenimentul E_i este critic, deci vîrful E_i se găsește pe un cel mai lung drum de la E_1 la E_n, fie $D = (E_1, \ldots, E_i, \ldots, E_n)$, a cărui lungime o notăm $l(D)$. În acest caz atît drumul $D_1 = (E_1, \ldots, E_i)$ format cu șirul de vîrfuri ale lui D dintre E_1 și E_i, cît și drumul $D_2 = (E_i, \ldots, E_n)$ format cu șirul de vîrfuri ale lui D dintre E_i și E_n, sînt drumuri de lungime maximă între extremitățile lor. Într-adevăr, dacă presupunem prin reducere la absurd că de exemplu D_1 nu are o lungime maximă, îl putem înlocui printr-un alt drum D_1' cu extremitățile E_1 și E_i, de lungime mai mare ca D_1. Însă în acest caz drumul D_1' împreună cu D_2 formează un drum D' de la E_1 la E_n, de lungime $l(D') = l(D_1') + l(D_2) > l(D_1) + l(D_2) = l(D)$, deci $l(D') > l(D)$. Am ajuns astfel la o contradicție, deoarece am făcut ipoteza că D este un drum de lungime maximă de la E_1 la E_n. Am demonstrat astfel că atît D_1 cît și D_2 sînt drumuri de lungime maximă în mulțimea drumurilor de aceleași extremități. Rezultă că $t_i = l(D_1)$ și $t_i^* = t_n - l(D_2)$. Deoarece $t_n = l(D) = l(D_1) + l(D_2)$, deducem că $t_i = t_i^* = l(D_1)$ pentru orice eveniment critic E_i dintr-un graf de activități. Am obținut astfel pentru fiecare eveniment două date:

— data t_i este *data așteptată* a realizării evenimentului E_i;

— data t_i^* este *data limită* de realizare a evenimentului E_i, după depășirea căreia timpul total de execuție a ansamblului lucrărilor este majorat.

Data t_i este numită și *data cea mai apropiată*, iar data limită t_i^* este numită și *data cea mai tîrzie* a evenimentului E_i.

Am văzut că pentru evenimentele critice data limită t_i^* se confundă cu data așteptată t_i. Intervalul de timp $[t_i, t_i^*]$ se numește *interval de fluctuație* și el reprezintă intervalul în care poate avea loc realizarea evenimentului necritic E_i, fără a modifica timpul total de execuție a ansamblului lucrărilor. În cazul evenimentelor critice acest interval se reduce la un singur număr, deoarece $t_i = t_i^*$.

Pentru exemplul considerat singurele evenimente necritice sînt C și F. Intervalul de fluctuație pentru C este $[6, 7]$, iar intervalul de fluctuație al evenimentului F este $[10, 11]$. Deci evenimentele C și F pot întîrzia cu cel mult o unitate de timp de la data lor așteptată, fără ca data evenimentului final t_H să se modifice. Această întîrziere poate avea loc de exemplu din cauza măririi timpilor operatori ai operațiilor (B, C), respectiv (D, F) cu cîte o unitate.

Dacă ambele operații (B, C) și (D, F) își măresc durata de la 3 la 4 unități, în graful de activități toate operațiile devin critice, iar toate cele 4 drumuri de la A la H devin drumuri critice. În acest caz nu ar mai fi admisă nici o întîrziere în executarea nici unei operații, pentru a nu produce întîrzierea întregii lucrări.

Este util să cunoaștem pentru fiecare operație o_{ij}, reprezentată de arcul (E_i, E_j) în graful de activități, întîrzierea care poate fi admisă la începerea sa, fără a modifica data așteptată de realizare a evenimentului E_j. Această întîrziere, numită *marginea liberă* a operației o_{ij}, este egală cu:

$$t_j - t_i - t_{ij}$$

unde t_i și t_j sînt datele așteptate ale evenimentelor E_i și E_j, care încadrează operația o_{ij} de durată t_{ij}.

Întîrzierea care poate fi admisă la începerea operației o_{ij}, fără a modifica data limită de realizare a evenimentului E_j, se numește *marginea totală* a operației o_{ij} și ea este egală cu

$$t_j^* - t_i - t_{ij}.$$

Aceasta este deci întîrzierea maximă care poate fi admisă la începerea operației o_{ij}, fără a mări durata de execuție a întregii lucrări.

Într-adevăr, operația o_{ij} are o durată egală cu t_{ij}, trebuie să se termine cel mai tîrziu la data t_j^* și nu poate începe mai devreme de data t_i, deci ea poate începe la cel mult $t_j^* - t_i - t_{ij}$ unități de timp după durata t_i. Deoarece $t_j^* \geqslant t_j$, rezultă că marginea totală este mai mare sau egală cu marginea liberă a oricărei operații.

Dacă operația o_{ij} se găsește pe drumul critic, deci este o operație critică, rezultă că evenimentele E_i și E_j sînt critice și în plus $t_j^* = t_j = t_i + t_{ij}$. Deci marginile libere, cît și marginile totale ale operațiilor critice sînt nule, începerea lor trebuind să se facă fără nici o întîrziere.

Intervalele de fluctuație și marginile operațiilor măsoară elasticitatea unei lucrări, în sensul că, cu cît acestea sînt mai reduse, cu atît termenele de execuție a diferitelor faze ale lucrării sînt mai rigide.

În cazul în care timpii operatori ai operațiilor necritice pot fi măriți, marginea liberă va corespunde măririi posibile a duratei unei operații necritice o_{ij}, lăsînd neschimbate datele t_i și t_j ale evenimentelor E_i și E_j care încadrează această operație.

Marginea totală a acestei operații va corespunde deci măririi maxime a duratei operației o_{ij}, astfel încît evenimentul E_j să poată avea loc la data sa limită de realizare t_j^*, după depășirea căreia toată lucrarea ar întîrzia, în condițiile în care evenimentul E_i se produce la data așteptată t_i.

De exemplu, pentru graful din figura III.5 marginea liberă a operației reprezentate de arcul (B, C) este

$$t_C - t_B - t_{BC} = 6 - 3 - 3 = 0.$$

în timp ce marginea sa totală este egală cu

$$t_C^* - t_B - t_{BC} = 7 - 3 - 3 = 1.$$

Deci operația (B, C) poate admite o întîrziere maximă de o unitate de timp, fără a produce întîrzierea execuției întregii lucrări.

Pentru operația necritică (D, F) marginea liberă este $t_F - t_D - t_{DF} = 0$ și marginea totală este egală cu $t_F^* - t_D - t_{DF} = 1$, iar pentru operația (F, G) marginea liberă este $t_G - t_F - t_{FG} = 1$ și marginea totală este $t_G^* - t_F - t_{FG} = 1$.

Cunoașterea intervalelor de fluctuație ale evenimentelor care compun o lucrare, precum și a marginilor totale ale operațiilor componente permite să se urmărească executarea la timp a întregii lucrări.

Astfel, dacă toate evenimentele se produc înaintea sau la datele lor limită t_i^*, lucrarea se va desfășura normal și data realizării finale t_n va fi respectată.

Dacă însă se va întîmpla ca data producerii unui eveniment E_i să depășească data sa limită t_i^*, pentru a nu se depăși data t_n de realizare a întregii lucrări va trebui să se accelereze execuția operațiilor situate pe cel mai lung drum de la E_i la E_n. Aceasta se poate face de exemplu prin alocarea de resurse suplimentare (forță de muncă, materiale sau utilaje) în detrimentul operațiilor progrmului care au margini totale mari.

Dacă nu există această posibilitate, se va putea evalua întîrzierea datei t_n de realizare a ansamblului operațiilor care compun lucrarea.

Metoda drumului critic este utilizată, din aceste motive, în urmărirea execuției unor proiecte de cercetare sau de dezvoltare care pot cuprinde mii de operații, în urmărirea lucrărilor de construcții-montaj a unor obiective economice etc.

III.2.3. **Determinarea drumului critic**

Să presupunem că graful de activități G are vîrfurile $x_1, \ldots, x_n$, unde x_1 reprezintă începerea lucrării, iar x_n reprezintă terminarea tuturor activităților lucrării. Vom nota prin t_{ij} durata activității reprezentate de arcul (x_i, x_j) al lui G.

Vom nota pentru orice vîrf x_i prin $p(x_i)$ predecesorii lui x_i, adică mulțimea vîrfurilor de la care pleacă arce care au extremitatea finală în x_i și prin $s(x_i)$ succesorii lui x_i, adică mulțimea vîrfurilor către care pleacă arce care au extremitatea inițială în x_i.

De exemplu, pentru graful de activități din figura III.6 obținem : $p(1) = \varnothing$; $s(1) = \{2, 3\}$, $p(4) = \{2, 3\}$, $s(4) = \{5, 6\}$, $p(6) = \{4, 5\}$, $s(6) = \varnothing$ etc.

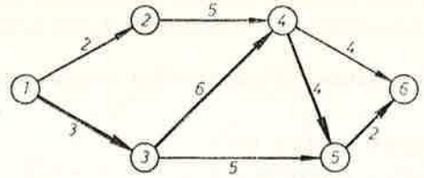


Fig. III.6

Vom arăta că datele t_i, respectiv t_i^* ale evenimentului reprezentat prin vîrful x_i al grafului G se pot determina printr-un calcul recursiv, bazat pe formulele :

a) $t_i = \max\limits_{x_j \in p(x_i)} (t_j + t_{ji})$,

unde toate datele t_j asociate vîrfurilor x_j care sînt extremități inițiale ale arcelor (x_j, x_i) au fost calculate la un pas anterior, plecînd de la $t_1 = 0$, care este data începerii lucrării.

b) $t_i^* = \min\limits_{x_j \in s(x_i)} (t_j^* - t_{ij})$,

unde toate datele t_j^* asociate vîrfurilor x_j care sînt extremități finale ale arcelor (x_i, x_j) au fost calculate la un pas anterior, plecîndu-se de la $t_n^* = t_n$, care este data terminării lucrării.

Într-adevăr, am văzut că t_i este lungimea maximă a drumurilor de la x_1 la x_i în graful G. Dacă x_j este penultimul vîrf al unui drum de lungime maximă de la x_1 la x_i, putem scrie $t_i = t_j + t_{ji}$, deoarece drumul considerat de lungime maximă de la x_1 la x_i se compune dintr-un drum de lungime maximă de la x_1 la x_j, de lungime t_j și din arcul (x_j, x_i) de lungime t_{ji}. Dacă însă $x_j \in p(x_i)$ și x_j nu este penultimul vîrf al nici unui drum de lungime maximă de la x_1 la x_i rezultă $t_i > t_j + t_{ji}$, de unde se deduce formula a).

Data limită t_i^* se obține scăzînd din data t_n a evenimentului final lungimea maximă a drumurilor de la x_i la x_n.

Fie $D = (x_i, x_j, \ldots, x_n)$ un drum de la x_i la x_n în graful G, unde $x_j \in s(x_i)$.

Să notăm prin D_j drumul care se obține din D prin suprimarea primului vîrf x_i. Este clar că D_j este un drum de la x_j la x_n în graful G și avem $l(D) = t_{ij} + l(D_j)$.

Putem deci scrie :

$$t_i^* = t_n - \max l(D) = t_n - \max_j (t_{ij} + \max_{D_j} l(D_j)) = \min_j (t_n - \max_{D_j} l(D_j) - t_{ij}) =$$

$$= \min_j (t_j^* - t_{ij}),$$

unde j parcurge mulțimea indicilor cu proprietatea că $x_j \in s(x_i)$, adică există un arc de la x_i la x_j, deoarece orice drum $D = (x_i, x_j, \ldots, x_n)$ de lungime maximă se compune din arcul (x_i, x_j) și un drum de lungime maximă de la x_j la x_n, iar $t_n - \max\limits_{D_j} l(D_j) = t_j^*$.

Să aplicăm aceste formule pentru graful de activități din figura III.6. Mai întîi vom calcula datele t_i cu formula a). Plecăm cu $t_1 = 0$. Deoarece $p(2) = \{1\}$ și $p(3) = \{1\}$, rezultă $t_2 = t_1 + t_{12} = 2$ și $t_3 = t_1 + t_{13} = 3$. Avem $p(4) = \{2, 3\}$, deci vom calcula

$$t_4 = \max(t_2 + t_{24}, t_3 + t_{34}) = \max(7, 9) = 9.$$

În continuare $p(5) = \{3, 4\}$, deci
$t_5 = \max(t_3 + t_{35}, t_4 + t_{45}) = \max(8, 13) = 13$.

În fine data terminării lucrării este

$t_0 = \max(t_4 + t_{40},\ t_5 + t_{50}) = \max(13,\ 15) = 15$, deoarece $p(6) = \{4,\ 5\}$.

Să calculăm acum datele limită t_i^* cu formulele b), plecînd de la evenimentul final către evenimentul inițial.

Avem $t_6^* = t_6 = 15$. Următorul vîrf care va fi luat în considerație este vîrful 5, deoarece $s(5) = \{6\}$. Găsim

$t_5^* = t_6^* - t_{56} = 13$. Apoi $s(4) = \{5,\ 6\}$, deci

$t_4^* = \min(t_6^* - t_{46},\ t_5^* - t_{45}) = \min(11,\ 9) = 9$.

În continuare putem alege fie vîrful 2, fie vîrful 3. De exemplu $t_2^* = t_4^* - t_{24} = 4$, deoarece $s(2) = \{4\}$. Apoi găsim:

$t_3^* = \min(t_4^* - t_{34},\ t_5^* - t_{35}) = \min(3,\ 8) = 3$.

Ca verificare obținem

$t_1^* = \min(t_3^* - t_{13},\ t_2^* - t_{12}) = \min(0,\ 2) = 0$.

Evenimentele pentru care $t_i = t_i^*$ vor fi tocmai evenimentele critice din graful activităților. În cazul nostru găsim că toate evenimentele cu excepția evenimentului 2, sînt critice.

Să observăm că dacă o activitate reprezentată de arcul $(x_i,\ x_j)$ are ambele evenimente care o delimitează critice, adică $t_i = t_i^*$ și $t_j = t_j^*$, nu rezultă că această activitate este critică, decît dacă $t_j = t_i + t_{ij}$.

De exemplu, activitatea (3, 5) din graful din figura III.6 are ambele extremități evenimente critice, însă nu este o operație critică.

Într-adevăr, ea are o margine totală egală cu $t_5^* - t_3 - t_{35} = 5$ unități, deci execuția ei poate admite o întîrziere de cel mult 5 unități de timp, fără a afecta data de realizare a întregii lucrări. În schimb operația (3, 4) este critică, deoarece $t_4 = t_3 + t_{34}$.

Deci activitățile critice ale grafului activităților se determină considerînd toate arcele cu ambele extremități evenimente critice și care în plus verifică condiția menționată.

Arcele critice compun drumul sau drumurile critice ale grafului activităților. În cazul exemplului considerat se obține un singur drum critic și anume (1, 3, 4, 5, 6). Cunoscînd datele t_i și t_i^* se pot determina intervalele de fluctuație ale evenimentelor, cît și marginile libere și totale ale activităților.

Se poate arăta că formulele a) și b) pot fi efectiv aplicate, deoarece graful activităților G nu conține circuite.

Dacă mulțimea A a vîrfurilor pentru care am determinat datele așteptate conține k elemente, pentru găsirea noii date t_i a vîrfului $x_i \notin A$ cu formula a) sînt necesare cel mult k adunări și $k - 1$ comparații, acest număr maxim fiind atins cînd $p(x_i) = A$.

Deci numărul de comparații necesare pentru a aplica formula a) este majorat de

$$\sum_{k=1}^{n-1} (k - 1) = \sum_{k=1}^{n-2} k = \frac{(n-1)(n-2)}{2},$$

iar numărul de adunări este majorat de

$$\sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}.$$

Obținem un rezultat analog pentru numărul de comparații și de scăderi necesitat de aplicarea formulei b). Deci pentru determinarea intervalelor de fluctuație ale celor n evenimente ale unui program prin acest algoritm sînt necesare în total cel mult $(n - 1)(n - 2)$ comparații și cel mult $n(n - 1)$ adunări și scăderi.

Probleme

III.2.1. Să se determine intervalele de fluctuație și drumul critic pentru graful de activități din figura III.7.

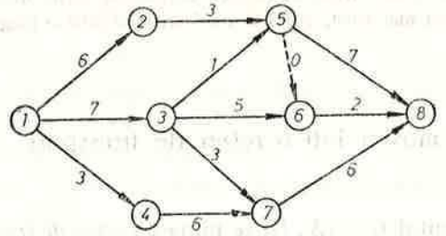


Fig. III.7

III.2.2. Să se construiască graful programului și să se determine drumul critic, intervalele de fluctuație ale evenimentelor și marginile totale ale operațiilor în următoarea situație de construcție a unui ansamblu hidroelectric, cu unitatea de timp luna. Operațiile care trebuie să fie realizate și duratele lor sînt următoarele:

a) Construcția drumurilor de acces la uzină și la carierele pentru obținerea materialelor ($t_a = 4$);
b) Pregătirea carierelor de exploatare și a fundațiilor ($t_b = 6$);
c) Construcția unui centru pentru personal și administrație ($t_c = 4$);
d) Comanda și construcția materialului electric și hidraulic (generatoare, turbine, conductă forțată etc.) ($t_d = 12$);
e) Construcția uzinei ($t_e = 10$);
f) Construcția barajului, a digurilor și a deversorului de suprafață ($t_f = 24$);
g) Construcția galeriilor de fugă și a conductelor de aducțiune ($t_g = 7$);
h) Montajul uzinei și al conductelor ($t_h = 10$);
i) Probele tehnologice ($t_i = 3$).

Evenimentele programului sînt următoarele:

1) Pornirea lucrărilor;
2) Terminarea drumurilor de acces;
3) Terminarea centrului pentru personal și administrație și a fundațiilor;
4) Terminarea uzinei și a montării conductelor de aducțiune;
5) Terminarea ansamblului energetic;
6) Livrarea lucrărilor către beneficiar.

III.2.3. Să se arate că orice graf orientat care nu conține circuite are cel puțin un vîrf x cu proprietatea $p(x) = \varnothing$ și cel puțin un vîrf y astfel încît $s(y) = \varnothing$.

III.2.4. Se consideră graful de activități din figura III.8, pentru care duratele operațiilor au fost scrise lîngă arcele respective.

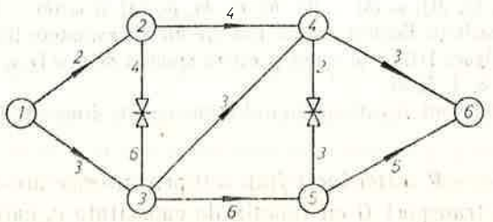


Fig. III.8

Operațiile reprezentate de arcul dintre vîrfurile 2 și 3 și arcul dintre vîrfurile 4 și 5 se numesc *operații disjunctive*, deoarece desenul din figură are următoarea interpretare: Există fie arcul (2, 3) de durată egală cu 4, fie arcul (3, 2) de durată egală cu 6. La fel pentru perechea de vîrfuri 4 și 5 există fie arcul (4, 5) de durată 2, fie arcul (5, 4) de durată 3, însă nu amîndouă aceste arce. Să se determine orientările arcelor disjunctive astfel încît întreaga lucrare să se termine într-un timp cît mai scurt, adică drumul critic să aibă o lungime minimă.

III.3. Flux maxim într-o rețea de transport

Un graf orientat $G = (X, U)$ se numește *rețea de transport* dacă satisface următoarele condiții:

a) există un vîrf unic $a \in X$ în care nu intră nici un arc sau $\omega^-(a) = \varnothing$, unde prin $\omega^-(a)$ se notează mulțimea arcelor care intră în vîrful a;

b) există un vîrf unic $b \in X$ din care nu iese nici un arc sau $\omega^+(b) = \varnothing$, unde prin $\omega^+(b)$ se notează mulțimea arcelor care ies din vîrful b;

c) G este conex și există drumuri de la a la b în G;

d) s-a definit o funcție $c: U \to \mathbf{R}$ astfel încît $c(u) \geqslant 0$ pentru orice arc $u \in U$.

Vîrful a se numește *intrarea rețelei*, vîrful b se numește *ieșirea rețelei*, iar numărul nenegativ $c(u)$ se numește *capacitatea* arcului u.

Un exemplu de rețea de transport cu 8 vîrfuri este reprezentat în figura III.9.

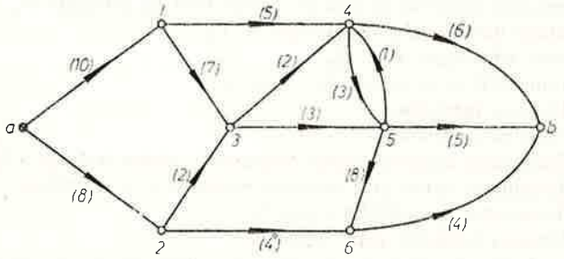


Fig. III.9

Vîrful a este intrarea rețelei, vîrful b este ieșirea rețelei și se verifică de exemplu că $\omega^-(a) = \varnothing$, $\omega^+(a) = \{(a, 1), (a, 2)\}$, $\omega^-(b) = \{(4, b), (5, b), (6, b)\}$ și $\omega^+(b) = \varnothing$.

Valorile capacității fiecărui arc au fost trecute în paranteze lîngă arcele respective.

Graful din figura III.9 este conex și există drumuri de la a la b, de exemplu (a, 1, 3, 5, b) sau (a, 2, 3, 4, 5, b).

Un exemplu de lanț de extremități a și b care nu este drum este [(a, 2), (2, 6), (5, 6), (5, b)].

O funcție $f: U \to \mathbf{R}$ astfel încît $f(u) \geqslant 0$ pentru orice arc u se numește *flux* în rețeaua de transport G cu funcția de capacitate c, care se notează $G = (X, U, c)$, dacă sînt îndeplinite următoarele două condiții:

C) Condiția de conservare a fluxului:

Pentru orice vîrf x cu $x \neq a$, $x \neq b$, suma fluxurilor pe arcele care intră în x este egală cu suma fluxurilor pe arcele care ies din x, adică:

$$\sum_{u \in \omega^-(x)} f(u) = \sum_{u \in \omega^+(x)} f(u)$$

pentru orice $x \in X \setminus \{a, b\}$.

Să observăm că această lege are aceeași formă cu prima lege a lui Kirchhoff din teoria rețelelor electrice.

M) Condiția de mărginire a fluxului:

Pentru orice arc al rețelei, valoarea fluxului nu poate depăși capacitatea arcului respectiv, adică

$$f(u) \leqslant c(u) \text{ pentru orice arc } u \in U.$$

Pentru rețeaua de transport din figura III.9 valorile unui flux pe arce au fost reprezentate în figura III.10, lîngă numărul din paranteze care reprezintă capacitatea arcului respectiv.

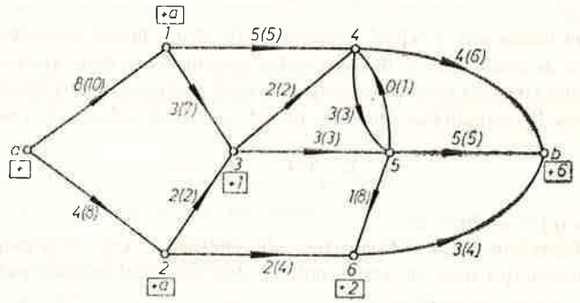


Fig. III.10

Se verifică condițiile *C* și *M* relative la vîrfurile și la arcele rețelei. De exemplu, pentru vîrful 4 condiția de conservare se scrie: $5 + 2 + 0 = 3 + 4$.

Pentru orice mulțime de vîrfuri $A \subset X$ vom defini o *tăietură de suport* A ca fiind mulțimea arcelor care intră în mulțimea A din exteriorul lui A și o vom nota astfel:

$$\omega^-(A) = \{(x, y) \mid x \notin A,\ y \in A \text{ și } (x, y) \in U\}.$$

Vom mai nota:

$$\omega^+(A) = \{(x, y) \mid x \in A,\ y \notin A \text{ și } (x, y) \in U\},$$

adică mulțimea arcelor care ies din mulțimea A de vîrfuri.

Capacitatea tăieturii $\omega^-(A)$ se notează prin $c(\omega^-(A))$ și ea este egală prin definiție cu $\sum_{u \in \omega^-(A)} c(u)$, adică cu suma capacităților arcelor care fac parte din tăietura considerată.

Pentru rețeaua de transport din figura III.9 dacă alegem $A = \{4, 3, 5, b\}$, obținem:

$$\omega^-(A) = \{(1, 4), (1, 3), (2, 3), (6, b)\} \text{ și}$$

$\omega^+(A) = \{(5, 6)\}$, iar capacitatea tăieturii de suport A este $c(\omega^-(A)) = 5 + 7 + 2 + 4 = 18$.

În continuare vom lucra numai cu tăieturi de această formă, avînd ca suport o mulțime de vîrfuri ale rețelei care conține ieșirea b și nu conține intrarea a.

Teorema III.3.1. *Fiind dată o rețea de transport cu intrarea a și ieșirea b și un flux f, are loc egalitatea:*

$$\sum_{u \in \omega^+(a)} f(u) = \sum_{u \in \omega^-(b)} f(u). \tag{1}$$

Demonstrație. Egalitatea (1) exprimă faptul că fluxul care iese din a ajunge în b, deoarece este verificată condiția de conservare în fiecare vîrf $x \neq a, b$. Să calculăm în două moduri suma:

$$\sum_{x \in X} \Big(\sum_{u \in \omega^-(x)} f(u) - \sum_{u \in \omega^+(x)} f(u) \Big), \tag{2}$$

adică pentru fiecare vîrf al rețelei facem diferența dintre fluxul pe arcele care intră în acel vîrf și fluxul pe arcele care ies din acel vîrf și apoi însumăm toate aceste diferențe.

Din cauza condiției C de conservare a fluxului pentru orice vîrf diferit de intrare și de ieșire, termenul corespunzător din suma (2) este nul. Deci suma (2) se reduce la

$$\sum_{u \in \omega^-(b)} f(u) - \sum_{u \in \omega^+(a)} f(u), \tag{3}$$

deoarece $\omega^-(a) = \omega^+(b) = \varnothing$.

Însă fiecare arc $(x, y) \in U$ aparține atît mulțimii $\omega^+(x)$, cît și mulțimii $\omega^-(y)$, cînd valoarea fluxului $f(u)$ apare cu semne contrare, deci regrupînd termenii putem scrie suma (2) sub forma

$$\sum_{u \in U} (f(u) - f(u)) = 0. \tag{4}$$

Comparînd (3) cu (4) rezultă (1).

În continuare vom numi valoarea comună a celor două sume care apar în egalitatea (1) *fluxul la ieșirea rețelei* (care deci este egal cu fluxul la intrarea rețelei) și îl vom nota cu f_b.

Pentru cazul fluxului reprezentat în figura III.10 fluxul la intrare este egal cu $8 + 4 = 12$, fluxul la ieșire este egal cu $f_b = 4 + 5 + 3 = 12$, iar capacitatea tăieturii de suport $A = \{4, 3, 5, b\}$ am văzut că este egală cu $18 > 12$.

Proprietatea fluxului la ieșire, de a fi mai mic sau egal cu capacitatea oricărei tăieturi are loc în orice rețea de transport, așa cum ne indică teorema următoare.

Teorema III.3.2. *Pentru o rețea de transport $G = (X, U, c)$ cu intrarea a și ieșirea b și un flux f, să considerăm o mulțime oarecare de vîrfuri $A \subset X$ cu proprietatea că $a \notin A$ și $b \in A$. Au loc relațiile:*

$$f_b = \sum_{u \in \omega^-(A)} f(u) - \sum_{u \in \omega^+(A)} f(u) \leqslant c(\omega^-(A)) \tag{5}$$

Demonstrație. Pentru a arăta că fluxul la ieșirea rețelei este egal cu diferența dintre suma fluxurilor pe arcele care intră în mulțimea A și suma fluxurilor pe arcele care ies din mulțimea de vîrfuri A, se poate proceda în mod analog ca pentru demonstrarea egalității (1), considerînd suma:

$$\sum_{x \in A} \Big(\sum_{u \in \omega^-(x)} f(u) - \sum_{u \in \omega^+(x)} f(u) \Big). \tag{6}$$

Deoarece pentru orice arc $u \in U$ există inegalitățile $0 \leqslant f(u) \leqslant c(u)$ putem scrie

$$f_b = \sum_{u \in \omega^-(A)} f(u) - \sum_{u \in \omega^+(A)} f(u) \leqslant \sum_{u \in \omega^-(A)} f(u) \leqslant \sum_{u \in \omega^-(A)} c(u) = c(\omega^-(A)).$$

În cele ce urmează vom studia problema determinării unui flux maxim f_b la ieșirea unei rețele de transport, pe care o vom numi pe scurt problema fluxului maxim.

Algoritmul lui Ford-Fulkerson pentru obținerea unui flux maxim.

Pentru obținerea unui flux maxim la ieșirea b a unei rețele de transport $G = (X, U, c)$, unde capacitatea $c(u) \geqslant 0$ a fiecărui arc este un număr întreg, se procedează după cum urmează:

1) Se pleacă de la un flux inițial care verifică condițiile de conservare în fiecare vîrf și de mărginire pe fiecare arc, de exemplu de la fluxul avînd componente nule pe fiecare arc al rețelei.

Deci putem lua $f(u) = 0$ pentru orice arc $u \in U$.

2) Se determină lanțurile nesaturate de la a la b (adică lanțurile pe care fluxul poate fi mărit). Se utilizează pentru aceasta următorul procedeu de etichetare:

a) Se marchează intrarea a cu $+$;

b) Un vîrf x fiind marcat, se va marca:

— cu $+x$ oricare vîrf y nemarcat cu proprietatea că arcul $u = (x, y)$ este nesaturat, adică $f(u) < c(u)$;

— cu $-x$ oricare vîrf y nemarcat cu proprietatea că arcul $u = (y, x)$ are un flux nenul, adică $f(u) > 0$.

Dacă prin acest procedeu de marcare se etichetează ieșirea b, atunci fluxul f_b obținut la pasul curent nu este maxim.

Se va considera un lanț format din vîrfuri etichetate (ale căror etichete au respectiv semnele $+$ sau $-$) care unește a cu b și care poate fi ușor găsit dacă se urmăresc etichetele vîrfurilor sale în sensul de la b către a.

Fie v acest lanț. Să notăm cu v^+ mulțimea arcelor (x, y) ale lui v, unde marcajul lui y are semnul $+$, deci care sînt orientate în sensul de la a către b și cu v^- mulțimea arcelor (x, y) ale lui v, unde marcajul lui y are semnul $-$, deci care sînt orientate în sensul de la b către a.

Calculăm $\varepsilon_1 = \min\limits_{u \in v^+} (c(u) - f(u))$, $\varepsilon_2 = \min\limits_{u \in v^-} f(u)$ și $\varepsilon = \min(\varepsilon_1, \varepsilon_2)$.

Din modul de etichetare rezultă $\varepsilon > 0$. Vom mări cu ε fluxul pe fiecare arc $u \in v^+$ și îl vom micșora cu ε pe fiecare arc $u \in v^-$, obținînd la ieșire un flux egal cu $f_b + \varepsilon > f_b$. Se repetă aplicarea pasului 2 cu fluxul nou obținut.

Dacă însă prin aplicarea pasului 2 nu mai putem marca ieșirea b, am obținut un flux maxim și ne oprim. În plus, mulțimea arcelor care unesc vîrfurile marcate cu vîrfurile nemarcate constituie o tăietură de capacitate minimă.

Din modul de definire a lui ε obținem un nou flux f' care verifică $0 \leqslant f'(u) \leqslant c(u)$ pentru orice arc $u \in U$. În plus, condiția de conservare în fiecare vîrf $x \neq a, b$ este în continuare verificată. Într-adevăr, dacă lanțul v trece prin x și cele două arce incidente cu x din v aparțin amîndouă fie lui v^+, fie lui v^-, se mărește cu ε (sau se micșorează cu ε) fluxul pe un arc de intrare și fluxul pe un arc de ieșire din x, deci condiția C este verificată pentru noul flux f'. Dacă cele două arce incidente cu x aparțin unul lui v^+ și altul lui v^-, atunci se micșorează, respectiv se mărește cu ε fluxul pe două arce, care sînt ambele sau arce de intrare în x sau arce de ieșire din x. Fluxul la ieșire crește deoarece $f'_b = f_b + \varepsilon > f_b$, ultimul arc al lanțului v care intră în b aparținînd mulțimii v^+.

În cazul rețelei de transport cu fluxul reprezentat în figura III.10, aplicînd procedeul de etichetare expus obținem că ieșirea b poate fi etichetată pe lanțul $[(a, 2), (2, 6), (6\ b)]$. Toate arcele acestui lanț au sensul de la a către b și etichetele vîrfurilor sînt trecute lîngă vîrfurile respective în figura III.10.

Găsim $\varepsilon = \varepsilon_1 = \min(8-4, 4-2, 4-3) = 1$, deci vom mări fluxul cu cîte o unitate pe arcele $(a, 2)$, $(2, 6)$, $(6, b)$, obținînd fluxul din figura III.11.

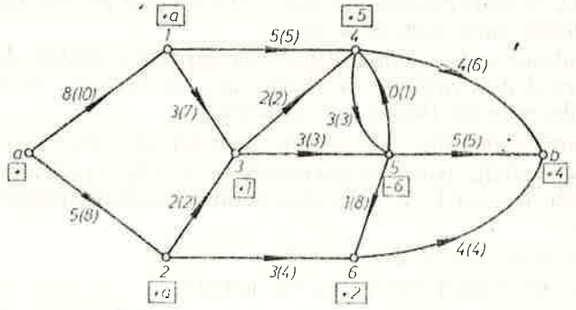


Fig. III.11

Ieșirea poate fi din nou etichetată, așa cum rezultă din figura III.11. Pentru a găsi lanțul nesaturat v de la a la b, procedăm astfel: b are marcajul $+4$, deci ultimul arc al lanțului v este $(4, b)$; vîrful 4 are eticheta $+5$, deci arcul anterior lui $(4, b)$ este $(5, 4)$ și nu $(4, 5)$ care are sensul de la b către a. Vîrful 5 are eticheta -6, deci următorul arc este $(5, 6)$; vîrful 6 are marcajul $+2$, deci următorul arc este $(2, 6)$; vîrful 2 are marcajul $+a$ și deci ultimul arc este $(a, 2)$. Scriind arcele obținute în ordinea inversă găsirii lor, deducem $v = [(a, 2), (2, 6), (6, 5), (5, 4), (4, b)]$. Calculăm $\varepsilon_1 = \min(8-5, 4-3, 1-0, 6-4) = 1$ și $\varepsilon_2 = \min(1) = 1$, deoarece $v^- = \{(5, 6)\}$.

Rezultă $\varepsilon = 1$, deci vom mări fluxul cu cîte o unitate pe arcele $(a, 2)$, $(2, 6)$, $(5, 4)$ și $(4, b)$ și îl vom micșora cu o unitate pe arcul $(5, 6)$, obținînd noul flux reprezentat în figura III.12.

Acum ieșirea nu mai poate fi etichetată, deci fluxul obținut este maxim.

Mulțimea vîrfurilor neetichetate este $A = \{4, 5, 6, b\}$, deci $\omega^-(A) = \{(1, 4), (3, 4), (3, 5), (2, 6)\}$, de capacitate $c(\omega^-(A)) = 14$. Arcele acestei tăieturi sînt traversate de curba închisă desenată în figura III.12 în jurul ieșirii b.

Fluxul la ieșire este $f_b = 5 + 5 + 4 = 14 = c(\omega^-(A))$. Conform teoremei III.3.2 pentru orice flux și orice tăietură de suport A există inegalitatea: $f_b \leqslant c(\omega^-(A))$. Deoarece am găsit un flux și o tăietură pentru care această inegalitate să fie chiar egalitate, rezultă că fluxul obținut în figura III.12 este maxim, iar tăietura $\omega^-(A)$ are o capacitate minimă.

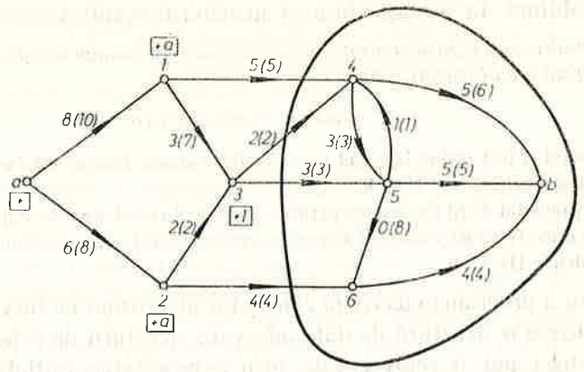


Fig. III.12

Teorema III.3.3. *Algoritmul lui Ford-Fulkerson are un număr finit de pași pentru orice rețea de transport cu capacități numere întregi. În momentul cînd prin procedeul de etichetare de la pasul 2) nu mai putem eticheta ieșirea rețelei, fluxul obținut este maxim, iar mulțimea arcelor care unesc vîrfurile etichetate cu vîrfurile care nu au putut fi etichetate formează o tăietură de capacitate minimă.*

Demonstrație. Vom arăta mai întîi că algoritmul are un număr finit de pași, majorat de exemplu de $\sum_{u \in \omega^-(b)} c(u)$, adică de suma capacităților arcelor care intră în b.

Într-adevăr, conform teoremei III.3.2 există inegalitatea $f_b \leqslant c(\omega^-(A))$, deci $\max f_b \leqslant c(\omega^-(A))$, unde $\omega^-(A)$ este o tăietură oarecare a rețelei. Dacă alegem $A = \{b\}$, rezultă că $\omega^-(A) = \omega^-(b)$. Obținem $\max f_b \leqslant c(\omega^-(A)) = \sum_{u \in \omega^-(b)} c(u)$.

Inițial se pleacă cu fluxul nul pe fiecare arc sau cu un flux oarecare, deci $f_b \geqslant 0$ și la fiecare pas fluxul f_b crește cu ε. Deoarece capacitățile arcelor sînt numere întregi și nenegative, rezultă că valorile fluxului pe arce și valorile lui ε sînt numere întregi, iar $\varepsilon > 0$ implică ε întreg, $\varepsilon \geqslant 1$.

Deci la fiecare pas fluxul f_b crește cu cel puțin o unitate, deci numărul de pași ai algoritmului este majorat de capacitatea unei tăieturi.

Să arătăm că, în momentul în care ieșirea nu mai poate fi marcată, fluxul obținut este maxim, iar mulțimea arcelor care unesc vîrfurile marcate cu vîrfurile nemarcate formează o tăietură de capacitate minimă. Pentru aceasta, să notăm cu A mulțimea vîrfurilor care nu pot fi etichetate cu algoritmul descris. Obținem $a \notin A$ și $b \in A$, deoarece am presupus că ieșirea rețelei nu poate fi marcată. Deci $\omega^-(A)$ formează o tăietură a rețelei de transport. Pentru orice arc $u \in \omega^-(A)$ obținem $f(u) = c(u)$, deoarece din $u = (x, y)$ rezultă că $x \notin A$ și $y \in A$. Dacă $f(u) < c(u)$ ar rezulta că vîrful y ar putea fi etichetat, ceea ce contrazice faptul că $y \in A$. Pentru orice arc $u \in \omega^+(A)$ obținem $f(u) = 0$, deoarece dacă $u = (y, x)$ rezultă că $y \in A$ și $x \notin A$. Dacă $f(u) > 0$, vîrful y ar putea fi etichetat plecînd de la vîrful etichetat x, ceea ce contrazice faptul că $y \in A$. Deci putem scrie, conform egalității din (5):

$$f_b = \sum_{u \in \omega^-(A)} f(u) - \sum_{u \in \omega^+(A)} f(u) = \sum_{u \in \omega^-(A)} c(u) = c(\omega^-(A)).$$

Însă pentru orice flux f și orice tăietură $\omega^-(A)$ existînd inegalitatea $f_b \leqslant c(\omega^-(A))$, rezultă că fluxul f_b este maxim, iar tăietura $\omega^-(A)$ are o capacitate minimă.

Am obținut în același timp și următorul rezultat :

Corolar. *Pentru orice rețea de transport, valoarea maximă a fluxului la ieșire este egală cu capacitatea minimă a unei tăieturi, adică* :

$$\max f_v = \min c(\omega^-(A)).$$

Acest rezultat a fost dedus în cazul capacităților întregi, însă el este valabil pentru orice funcție de capacitate $c: U \to \mathbf{R}_+$.

Algoritmul lui Ford-Fulkerson permite găsirea fluxului maxim după un număr finit de pași în orice rețea de transport G pentru care capacitățile arcelor sînt numere raționale (vezi problema III.3.2.).

Pentru a programa efectiv la calculator algoritmul de flux maxim trebuie să se utilizeze o structură de date adecvată structurii de rețea. De exemplu, arcele rețelei pot fi reprezentate prin extremitatea inițială, extremitatea finală, capacitatea și fluxul lor în această ordine sub forma unei liste liniare. O altă listă poate memora marcajele tuturor vîrfurilor diferite de intrarea rețelei :

De exemplu, pentru rețeaua din figura III.11 această reprezentare poate arăta astfel :

L_1 | a | 1 | 10 | 8 | a | 2 | 8 | 5 | 1 | 3 | 7 | 3 | ...

(a, 1) (a, 2) (1, 3)

unde primele patru poziții ale listei L_1 memorează informația relativă la arcul (a, 1) în ordinea menționată, următoarele 4 cea relativă la arcul (a, 2) următoarele 4 cea relativă la arcul (1, 3) ș.a.m.d.

Pentru a ușura găsirea arcelor care pleacă din fiecare vîrf se poate folosi o altă listă L_2 care indică pe care poziție a listei L_1 încep arcele care pleacă din fiecare vîrf diferit de ieșire.

Pentru exemplul considerat, vom avea L_2 | 1 | 9 | 17 | ... deoarece reprezentarea

a 1 2

arcelor care pleacă din a începe în poziția 1 a listei L_1, reprezentarea arcelor care pleacă din 1 începe în poziția 9 a listei L_1, a arcelor care pleacă din 2 în poziția 17 ș.a.m.d.

Folosirea listei L_2 asociată cu L_1 face de fapt inutilă reprezentarea în lista L_1 a extremităților inițiale : a, 1, 2, ..., ale arcelor rețelei G. Pentru a ușura găsirea arcelor care intră într-un vîrf x, ceea ce este necesar pentru procesul de marcare, se poate utiliza o listă L_1' și o listă L_2' asociată care să memoreze arcele care intră în vîrful 1, vîrful 2, ..., în b, în această ordine.

În fine, lista pentru marcaj în cazul rețelei din figura III.11 arată astfel :

L_3 | $+a$ | $+a$ | $+1$ | $+5$ | -6 | $+2$ | $+4$ |

Poziția întîi din L_3 este rezervată pentru marcajul vîrfului 1, poziția a II-a, pentru marcajul vîrfului 2, ..., ultima poziție pentru marcajul ieșirii b. Plecînd de la lista L_3 se poate găsi simplu lanțul nesaturat care permite creșterea fluxului la ieșire : Ultimul marcaj din L_3 este $+4$, deci ultimul arc este (4, b). Pe poziția a IV-a în L_3 găsim $+5$, deci arcul anterior este (5, 4). Pe poziția 5 găsim în L_3 numărul -6. Acest număr fiind negativ, arcul anterior nu este (6, 5), ci este (5, 6) ș.a.m.d.

În cazul rețelei din figura III.12 lista L_3 arată astfel :

L_3 | $+a$ | $+a$ | $+1$ | 0 | 0 | 0 | 0 |

deoarece vîrfurile 4, 5, 6 și b nu au putut fi marcate.

Aplicații

Unele probleme de transport sînt probleme de flux. De exemplu să presupunem că există n depozite conținînd respectiv cantitățile $a_1, \ldots, a_n$ dintr-un anumit produs, care trebuie transportate la m destinatari, care pot primi cel mult $b_1, \ldots, b_m$ unități din produs. Problema maximizării cantității totale care poate fi transportată se reduce la problema găsirii unui flux maxim în rețeaua de transport definită astfel : Graful G are vîrfurile $x_1, \ldots, x_n$; $y_1, \ldots, y_m$, o intrare a și o ieșire b. Mulțimea U este formată din : arcele (a, x_i) cu o capacitate egală cu a_i pentru $1 \leqslant i \leqslant n$; arcele (y_j, b) cu o capacitate egală cu b_j pentru $1 \leqslant j \leqslant m$ și toate arcele de forma (x_i, y_j) pentru $1 \leqslant i \leqslant n$ și $1 \leqslant j \leqslant m$, care au o capacitate corespunzătoare capacității de transport de la x_i la y_j (problema III.3.3).

Vom putea satisface toate cererile în $y_1, \ldots, y_m$ dacă oferta este mai mare sau egală cu cererea, ceea ce se scrie $\sum_{i=1}^{n} a_i \geqslant \sum_{j=1}^{m} b_j$ și dacă toate capacitățile de transport sînt suficient de mari. Determinarea tăieturii de capacitate minimă prezintă interes într-o serie de aplicații (vezi problemele III.3.5 și III.3.6).

Probleme

III.3.1. Să se găsească fluxul maxim în rețeaua de transport din figura III.13.

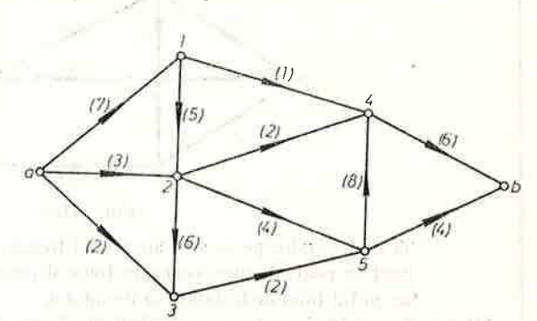


Fig. III.13

III.3.2. Aceeași problemă pentru rețeaua de transport cu capacități numere fracționare din figura III.14.

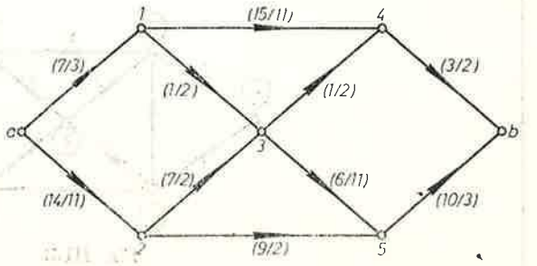


Fig. III.14

III.3.3. Să se găsească planul de transport pentru a transporta o cantitate maximă de produse din x_1, x_2, x_3 în y_1, y_2, y_3, știind că se pot transporta din x_i în y_j următoarele cantități maxime, date în tabelul :

	y_1	y_2	y_3
x_1	12	21	46
x_2	15	18	34
x_3	9	26	12

La intersecția liniei x_i cu coloana y_j se găsește cantitatea de produse care pot fi transportate din x_i în y_j.

Disponibilul la centrele x_i este dat de $a_1 = 60$, $a_2 = 24$ și $a_3 = 36$, iar cererile în centrele y_j sînt următoarele : $b_1 = 33$, $b_2 = 19$, $b_3 = 68$. Să se transforme această problemă într-o problemă de flux maxim într-o rețea de transport și să se determine un plan optim de transport utilizînd algoritmul lui Ford-Fulkerson.

III.3.4. Fie G o rețea de transport cu intrarea a, ieșirea b și $\omega^-(A)$ o tăietură oarecare cu $a \notin A$ și $b \in A$. Să se arate că orice drum $D=(a, \ldots, b)$ de la a la b în rețea conține cel puțin două vîrfuri vecine x_i și x_{i+1}, astfel încît arcul $(x_i, x_{i+1}) \in \omega^-(A)$.

III.3.5. Două localități a și b sînt legate printr-o rețea de drumuri cu 6 puncte de intersecție. Pentru a ajunge din a în b trebuie să urmăm drumurile de la a la b din graful din figura III.15. Costurile de instalare a unor posturi de control al circulației pe arcele acestui graf sînt indicate prin numerele scrise în paranteze lîngă arcele respective.

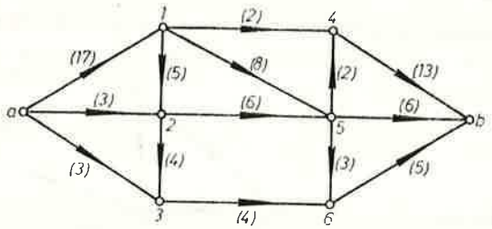


Fig. III.15

Să se determine pe ce arce ale rețelei trebuie instalate posturile de control, astfel încît să poată fi supravegheate toate drumurile posibile de deplasare din a în b, iar costul total de instalare să fie minim.

III.3.6. Se consideră graful de activități din figura III.16, unde vîrful 1 reprezintă începerea lucrării, vîrful 7 reprezintă terminarea ei, iar timpii operatori sînt numerele scrise în dreptul fiecărui arc.

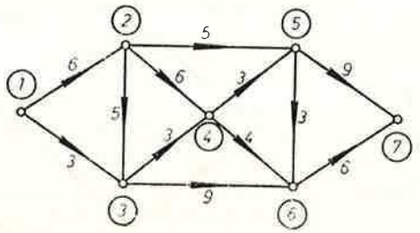


Fig. III.16

Costul necesar pentru a determina accelerarea operațiilor programului cu cîte o unitate de timp este trecut în tabelul următor :

Operația	Costul	Operația	Costul
(1,2)	6	(3,6)	3
(1,3)	2	(4,5)	4
(2,3)	6	(4,6)	1
(2,4)	1	(5,6)	1
(2,5)	3	(5,7)	1
(3,4)	5	(6,7)	5

Să se determine care operații trebuie să fie accelerate cu cîte o unitate de timp, astfel încît durata de execuție a întregii lucrări să se scurteze cu o unitate de timp și costul total pentru scurtarea duratelor acestor operații să fie minim.

IV. ALGORITMI ŞI METODE DE REPREZENTARE

IV.1. Noţiunea de algoritm

IV.1.1. **Definiţia algoritmului**

Noţiunea de algoritm este o noţiune matematică foarte veche. Cuvîntul „algoritm" este de origine arabă. El derivă din numele matematicianului „Abu Ja'far Mohammed ibn Mûsâ al Horezmi" care a scris o carte celebră intitulată „Kitab al jabr w'al-muquabala". Din titlul acestei cărţi provine cuvîntul algebră.

În evul mediu se folosea termenul de „algorism" cu înţelesul de proces al efectuării operaţiilor aritmetice cu ajutorul cifrelor arabe. Se presupune că din asocierea cuvîntului algorism cu domeniul lui de referinţă, aritmetica, a rezultat termenul algoritm. Începînd cu anul 1950 în toate manualele de specialitate cuvîntul algoritm este frecvent asociat cu procesul de aflare a celui mai mare divizor comun a două numere naturale, aşa-numitul „algoritm al lui Euclid". De asemenea, regulile operaţiilor aritmetice sînt denumite algoritmi de efectuare a operaţiilor respective.

Noţiunea de algoritm nu are o definiţie matematică. În aceeaşi situaţie se află şi alte noţiuni din matematică, cum ar fi noţiunea de mulţime.

Prin *algoritm* se acceptă să se înţeleagă un sistem de calcule, care, pentru o anumită clasă de probleme, din condiţiile iniţiale ale problemei permite să se obţină soluţia problemei respective, cu ajutorul unui şir finit şi ordonat de operaţii univoc determinate, efectuate mecanic, fără aportul creator al omului.

Există totuşi definiţii matematice riguroase pentru unele categorii de algoritmi : funcţii recursive, maşini Türing, algoritmii normali ai lui A. A. Markov. S-a demonstrat că aceste clase de algoritmi sînt echivalente. S-a emis ipoteza că oricare dintre aceste clase defineşte noţiunea de algoritm. Pînă în prezent această ipoteză nu a fost infirmată, deoarece nu s-a găsit nici un algoritm care să nu poată fi reprezentat conform regulilor acestor clase.

Un algoritm este compus din unul sau mai mulţi paşi, un pas reprezentînd efectuarea unei singure operaţii din şirul celor care alcătuiesc algoritmul.

Exemple

1. Algoritmul împărţirii întregi a două numere naturale

Se ştie că împărţirea întreagă a două numere constă din efectuarea unor scăderi succesive, pînă cînd descăzutul devine mai mic decît scăzătorul. Pentru fiecare scădere care se efectuează, descăzutul este rezultatul scăderii precedente, iar scăzătorul este împărţitorul. Rezultatul ultimei scăderi efectuate este tocmai restul împărţirii celor două numere, iar numărul de scăderi efectuate reprezintă cîtul împărţirii.

Paşii acestui algoritm sînt constituiţi de operaţiile de scădere şi de operaţiile de comparare a descăzutului cu scăzătorul. Este evident că şirul acestor operaţii este finit, deoarece descăzutul se micşorează cu fiecare nouă scădere, în timp ce scăzătorul rămîne neschimbat.

Fie, de exemplu, numerele 17 şi 7. Paşii algoritmului care duc la aflarea cîtului şi restului împărţirii sînt prezentaţi în tabelul IV.1.

Operaţia	Pasul	Numărul scăderii
scădere	$17 - 7 = 10$	1
comparare	$10 < 7$ nu	—
scădere	$10 - 7 = \boxed{3}$	2
comparare	$3 < 7$ da	—
	(descăzutul este mai mic decît scăzătorul)	

Tabelul IV.1

Numărul de scăderi efectuate este 2, iar rezultatul ultimei scăderi efectuate este 3, deci cîtul împărţirii numărului 17 prin 7 este 2, iar restul este 3.

2. Algoritmul lui Euclid

Acest algoritm se foloseşte pentru obţinerea celui mai mare divizor comun a două numere naturale.

Notînd cele două numere naturale prin m şi n, vom presupune că m este mai mare decît n.

Algoritmul constă din efectuarea unui şir de împărţiri întregi pînă cînd se obţine un rest nul. Pentru fiecare împărţire care se efectuează, împărţitorul este restul împărţirii precedente, iar deîmpărţitul este împărţitorul din împărţirea precedentă. Împărţitorul din ultima împărţire efectuată constituie cel mai mare divizor comun al celor două numere.

Paşii acestui algoritm sînt constituiţi de operaţiile de împărţire şi de verificare a anulării restului. Deoarece restul unei împărţiri este mai mic decît împărţitorul, şirul de resturi al împărţirilor succesive este strict descrescător, astfel că numărul de împărţiri din algoritm este finit.

Fie, de exemplu, numerele 78 şi 30. Paşii algoritmului care conduc la aflarea celui mai mare divizor comun al acestor numere sînt prezentaţi în tabelul IV.2.

Pasul	Operaţia
$78 : 30 = 2$ rest 18	împărţire
$18 = 0$ nu	verificare
$30 : 18 = 1$ rest 12	împărţire
$12 = 0$ nu	verificare
$18 : 12 = 1$ rest 6	împărţire
$6 = 0$ nu	verificare
$12 : \boxed{6} = 2$ rest 0	împărţire
$0 = 0$ da	verificare

Tabelul IV.2

IV.1.2. **Proprietăţile algoritmilor**

Orice algoritm trebuie să se bucure de următoarele trei proprietăţi fundamentale, numite proprietăţi de bază ale algoritmilor.

1. Claritatea. Orice algoritm trebuie să fie caracterizat printr-o descriere precisă, riguroasă, fără ambiguităţi a tuturor acţiunilor care urmează să se execute. Cu alte cuvinte, un algoritm, datorită caracterului său de automatism, trebuie să precizeze în mod univoc toate etapele de calcul pe care le va urma executantul algoritmului (omul sau maşina). De aceea, această proprietate este denumită uneori şi unicitate.

2. Generalitatea. Un algoritm este util dacă rezolvă nu numai o problemă particulară, concretă, ci o întreagă clasă de probleme asemănătoare. Aceasta înseamnă că un algoritm trebuie să se aplice la o mulţime de sisteme de date iniţiale. Această mulţime poartă numele de domeniu de aplicabilitate al algoritmului.

De exemplu, algoritmul lui Euclid se poate aplica la orice pereche de numere naturale. Vom spune deci că domeniul de aplicabilitate al algoritmului lui Euclid este mulţimea perechilor de numere naturale. Această proprietate este cunoscută şi sub numele de universalitate.

3. Eficacitatea. Orice algoritm urmăreşte prin execuţia sa obţinerea unui anumit rezultat. Pentru aceasta nu este suficient ca acţiunile algoritmului să fie bine determinate, ci trebuie ca pentru orice sistem de date iniţiale numărul de acţiuni (paşi) care urmează să se execute să fie finit. De aceea, această proprietate poartă denumirea de finitudine. La exemplele de algoritmi prezentate anterior s-a arătat că numărul de paşi corespunzători unui sistem oarecare de date iniţiale este finit.

IV.1.3. **Structura algoritmilor**

Acţiunile componente ale unui algoritm se efectuează asupra unor date iniţiale sau asupra unor rezultate intermediare ale operaţiilor anterioare. Atît datele cît şi rezultatele intermediare apar ca valori ale unor variabile. O variabilă are, în cadrul unui algoritm, o semnificaţie deosebită de aceea din matematică. Astfel, în timp ce în matematică o variabilă reprezintă o nedeterminată cu care se pot face operaţii matematice fără a fi cunoscută valoarea sa, într-un algoritm variabilele sînt utilizate pentru a denumi date sau rezultate intermediare. Deci, o variabilă este destinată să aibă o anumită valoare. Această valoare poate fi totuşi schimbată pe parcursul algoritmului. Este cazul aşa-numitelor variabile de lucru. O variabilă de lucru este folosită pentru reţinerea unui rezultat intermediar şi poate fi refolosită pentru reţinerea altui rezultat intermediar, atunci cînd rezultatul anterior nu mai este necesar pentru alte calcule.

Acţiunile unui algoritm se realizează sub forma unor operaţii. Aceste operaţii constituie paşii algoritmului.

Operaţiile care pot apărea într-un algoritm sînt de două categorii : operaţii de calcul şi operaţii de decizie.

O operaţie de calcul constă în efectuarea calculelor indicate de o expresie simbolică, înlocuind fiecare variabilă cu valoarea sa. Razultatul acestor calcule adică valoarea expresiei respective, este reţinut sub forma valorii unei variabile. Se spune, de obicei, că valoarea expresiei este atribuită variabilei respective.

Notaţia utilizată pentru operaţiile de calcul poate fi dedusă din următoarele exemple :

$$x \leftarrow y^2 + 2$$

$$\alpha \leftarrow \sqrt{\beta^2 - 1/2}$$

$$S \leftarrow LUNGIME \times LĂŢIME$$

$$X_0 \leftarrow \frac{x_1 + x_2}{2}.$$

Observaţii

1. Variabilele pot fi notate atît prin litere, cît şi prin cuvinte. Se pot utiliza şi indici.
2. Simbolul „←", care se citeşte „ia valoarea", este întrebuinţat pentru a marca atribuirea unei valori variabilei indicate. În matematică pentru acelaşi scop se utilizează simbolul „=". Simbolul „=" este însă folosit şi pentru relaţia de egalitate. Pentru a evita ambiguitatea în cadrul unui algoritm, se preferă folosirea a două simboluri distincte, astfel că simbolul „=" va fi folosit numai pentru a marca relaţia de egalitate.

Într-o operaţie de calcul se poate atribui o nouă valoare unei variabile a cărei valoare este utilizată în cadrul calculului respectiv.

De exemplu, operaţia de calcul

$$X \leftarrow X^2$$

atribuie variabilei X o nouă valoare, egală cu vechea valoare ridicată la pătrat. În acest fel iese în evidenţă modul de succesiune a valorilor unei variabile. Astfel, dacă înainte de efectuarea operaţiei de atribuire prezentate, variabila X are valoarea 3, după efectuarea operaţiei, aceeaşi variabilă va avea valoarea 9.

Un rol deosebit de important în structura unui algoritm îl au operaţiile de decizie.

În general, prin operaţie de decizie se înţelege determinarea valorii logice de adevăr a unei propoziţii. Propoziţiile analizate de operaţiile de decizie sînt propoziţii enunţiative, care nu pot fi decît adevărate sau false. De obicei, aceste propoziţii enunţă că un obiect are o anumită proprietate (de exemplu : valoarea variabilei X este pozitivă, variabila Y are ca valoare un număr întreg). De cele mai multe ori, propoziţiile asupra cărora se aplică operaţia de decizie se referă la o relaţie între două obiecte. De exemplu : valoarea variabilei X este egală cu valoarea variabilei Y, valoarea variabilei X se divide prin valoarea variabilei Y etc.

Rezultatul unei operaţii de decizie îl constituie valoarea „adevărat" sau „fals" a propoziţiei analizate. În cadrul acestei operaţii se calculează valorile diferitelor expresii care constituie obiectele relaţiilor respective, ţinînd cont de valorile variabilelor care apar în aceste expresii. Astfel, propoziţia

$$X + 2 > Y - 1$$

are valoarea logică „adevărat" dacă, de exemplu, variabila X are valoarea 7, iar variabila Y valoarea 2 şi valoarea logică „fals" dacă variabilele X şi Y au valorile 3 şi, respectiv, 8.

O importanţă deosebită în descrierea unui algoritm o are şi specificarea succesiunii de efectuare a operaţiilor componente.

Înlănţuirea paşilor unui algoritm poate fi indicată implicit sau explicit.

Astfel, succesiunea implicită de efectuare a paşilor unui algoritm este dată de ordinea de prezentare a acestor paşi în cadrul algoritmului.

Specificarea explicită a succesiunii de efectuare a unor pași apare în cazul operațiilor de decizie. În acest caz se specifică (explicit) care pas urmează să fie executat dacă rezultatul operației de decizie este „adevărat" și care pas dacă rezultatul operației este „fals". Deoarece acești doi pași următori sînt obligatoriu diferiți, operațiile de decizie reprezintă ramificații în succesiunea de pași ai unui algoritm. Să exemplificăm cele două tipuri de operații în cadrul algoritmului împărțirii întregi (cu rest).

Să notăm variabilele acestui algoritm astfel :

D — deîmpărțitul ;
I — împărțitorul ;
C — cîtul ;
R — restul.

Presupunînd că variabilele D și I au deja valorile corespunzătoare datelor algoritmului, pașii acestuia sînt :

Pasul 1. $C \leftarrow 0$;
Pasul 2. $R \leftarrow D$;
Pasul 3. $R < I$; dacă este adevărat urmează pasul 7,
dacă este fals urmează pasul 4 ;
Pasul 4. $R \leftarrow R - I$;
Pasul 5. $C \leftarrow C + 1$;
Pasul 6. $R < I$; dacă este adevărat urmează pasul 7,
dacă este fals urmează pasul 4 ;

Pasul 7. Terminarea algoritmului.

Se observă că pașii 1, 2, 4 și 5 specifică operațiile de calcul, în timp ce pașii 3 și 6 reprezintă operații de decizie. În tabelul IV.3 se prezintă succesiunea pașilor algoritmului în cazul numerelor 17 și 7.

Pasul	Valorile variabilelor înainte de execuția pasului				Rezultatul operației de decizie
	D	I	C	R	
1	[17]	[7]	—	—	—
2	17	7	0	—	—
3	17	7	0	17	fals
4	17	7	0	17	—
5	17	7	0	10	—
6	17	7	1	10	fals
4	17	7	1	10	—
5	17	7	1	3	—
6	17	7	2	3	adevărat
7	17	7	[2]	[3]	—

Tabelul IV.3

Este posibil ca să se specifice explicit următorul pas și în cazul operațiilor de calcul. Astfel, în exemplul anterior, observînd identitatea pașilor 3 și 6, se poate elimina pasul 6, specificînd la pasul 5 că urmează pasul 3.

Rezultă astfel următorul algoritm :

Pasul 1. $C \leftarrow 0$;
Pasul 2. $R \leftarrow D$;

Pasul 3. $R < I$, dacă este adevărat urmează pasul 6,
dacă este fals urmează pasul 4 ;
Pasul 4. $R \leftarrow R - I$;
Pasul 5. $C \leftarrow C + 1$, urmează pasul 3 ;
Pasul 6. Terminarea algoritmului.

De asemenea, este posibilă specificarea implicită a pasului următor și în cazul operațiilor de decizie. Totuși, deoarece într-un pas corespunzător unei operații de decizie se specifică doi pași următori, rămîne necesară specificarea explicită a unuia dintre acești doi pași. Astfel, în exemplul prezentat poate fi modificat pasul 3 în modul următor :

Pasul 3. dacă $R < I$ urmează pasul 6.

IV.1.4. Clasificarea algoritmilor

În funcție de structura lor algoritmii pot fi împărțiți în mai multe clase.

Algoritmii liniari sînt acei algoritmi care sînt alcătuiți numai din operații de calcul. Absența operațiilor de decizie din cadrul algoritmilor liniari are ca efect execuția pașilor acestor algoritmi într-o singură succesiune. În această categorie intră, de exemplu, algoritmul pentru calculul valorii unei expresii, cum ar fi algoritmul pentru calculul valorii polinomului $ax^2 + bx + c$, prezentat în continuare.

Pasul 1. $v \leftarrow a$;
Pasul 2. $v \leftarrow v \times x + b$;
Pasul 3. $v \leftarrow v \times x + c$;
Pasul 4. Terminarea algoritmului.

Algoritmii liniari sînt cei mai simpli algoritmi.

Algoritmii cu ramificații reprezintă acei algoritmi care cuprind și operații de decizie printre operațiile de calcul. În acest caz, în funcție de valorile variabilelor și de rezultatele operațiilor de decizie, pentru un algoritm cu ramificații există mai multe posibilități în ceea ce privește ordinea de execuție a pașilor săi.

Algoritmii aciclici cuprind acea categorie de algoritmi cu ramificații, pentru care în cadrul execuției nu se poate efectua de mai multe ori un același pas. Algoritmul rezolvării unei ecuații de gradul doi, $ax^2 + bx + c = 0$, este un exemplu de astfel de algoritm.

Pasul 1. DELTA $\leftarrow b^2 - 4ac$;
Pasul 2. dacă DELTA < 0 urmează pasul 5 ;
Pasul 3. $x_1 \leftarrow \dfrac{-b + \sqrt{\text{DELTA}}}{2a}$;
Pasul 4. $x_2 \leftarrow \dfrac{-b - \sqrt{\text{DELTA}}}{2a}$; urmează pasul 7 ;
Pasul 5. $\text{RE} \leftarrow \dfrac{-b}{2a}$;
Pasul 6. $\text{IM} \leftarrow \dfrac{\sqrt{-\text{DELTA}}}{2a}$;
Pasul 7. Terminarea algoritmului.

Se observă că nu sînt posibile decît două succesiuni de pași și anume : 1, 2, 3, 4, 7 în cazul rădăcinilor reale și 1, 2, 5, 6, 7 în cazul rădăcinilor complexe.

Algoritmii pentru care există posibilitatea repetării execuției unuia sau mai multor pași, alcătuiesc categoria *algoritmilor ciclici*. Succesiunea de pași care poate fi executată în mod repetat poartă denumirea de *ciclu*. Numărul de repetări ale unui ciclu poate fi fix sau variabil.

Cea mai mare parte a algoritmilor utilizați în practică este reprezentată de algoritmii ciclici. Acești algoritmi permit o descriere concisă a unor succesiuni cu numeroși pași. Algoritmii ciclici sînt indicați în special pentru a fi executați cu ajutorul calculatoarelor electronice. În acest caz numărul mare de pași de efectuat este compensat de viteza mare de execuție a operațiilor.

Exemplele de algoritmi prezentate la început, algoritmul lui Euclid și algoritmul împărțirii întregi constituie algoritmi ciclici. Ciclul algoritmului lui Euclid este reprezentat de pașii corespunzători efectuării unei împărțiri, iar ciclul algoritmului împărțirii întregi este format din pașii corespunzători efectuării unei scăderi.

Un algoritm ciclic poate avea mai multe cicluri, acestea putînd fi incluse unul în altul.

IV.1.5. **Exerciții**

1. Fie N un număr natural. Este corect următorul algoritm:
   Pasul 1. $M \leftarrow 2N + 1$
   Pasul 2. Dacă $M = 0$ atunci urmează pasul 4;
   Pasul 3. $M \leftarrow M - 2$, urmează pasul 2;
   Pasul 4. Terminarea algoritmului.

2. Se dă algoritmul:
   Pasul 1. $I \leftarrow 1$;
   Pasul 2. $S \leftarrow 1$;
   Pasul 3. $S \leftarrow S \times I^2$;
   Pasul 4. $I \leftarrow I + 2$;
   Pasul 5. Dacă $I \leqslant 5$ atunci urmează pasul 3;
   Pasul 6. Terminarea algoritmului.
   Care este valoarea lui S la terminarea algoritmului?

3. Să se scrie pașii algoritmului lui Euclid pentru numerele 459 și 170.

4. Se dă polinomul $P(x) = a_0x^4 + a_1x^3 + a_2x^2 + a_3x + a_4$. Să se alcătuiască un algoritm care să calculeze valoarea $P(x_0)$.

5. Să se găsească un algoritm care să determine cel mai mare număr dintre trei numere date.

6. Folosind schema lui Horner să se conceapă un algoritm care să calculeze cîtul și restul împărțirii polinomului $ax^3 + bx^2 + cx + d$ la $x - x_0$.

7. Să se scrie un algoritm pentru calculul sumei primilor n termeni ai unei progresii geometrice cu rația q și primul termen a.

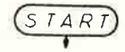
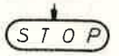
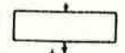
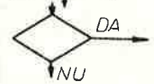
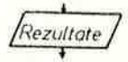
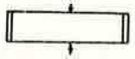
8. Să se conceapă un algoritm pentru calculul numărului de combinări C_n^k.

IV.2. Metode de reprezentare a algoritmilor

IV.2.1. **Scheme logice**

Cea mai simplă metodă de reprezentare a algoritmilor este metoda folosită în exemplele anterioare, care utilizează pentru descrierea fiecărui pas al algoritmului un limbaj apropiat de cel obișnuit.

Schemele logice constituie forme grafice mai sugestive de reprezentare a algoritmilor, cu o răspîndire largă, o schemă logică fiind alcătuită din blocuri între care se stabilesc legături orientate (săgeți). Blocurile au diferite forme grafice, în funcție de acțiunile (pașii) pe care le (îi) reprezintă (tabelul IV.4).

Denumirea blocului	Simbolul
Terminal	 
De calcul	
De decizie	
De intrare	
De ieșire	
De procedură	

Tabelul IV.4

Operațiile de calcul sînt reprezentate prin blocuri de calcul, iar operațiile de decizie prin blocuri de decizie.

Pentru a pune în evidență începutul sau sfîrșitul unei scheme logice se utilizează blocuri terminale.

Datele de intrare ale algoritmului precum și rezultatele acestuia sînt puse în evidență cu ajutorul unor blocuri de intrare-ieșire.

Legăturile care se stabilesc între blocuri reprezintă succesiunea efectuării acțiunilor reprezentate de blocurile respective.

Se remarcă, la blocurile de decizie, legăturile corespunzătoare celor două rezultate posibile ale operației de decizie.

Schema logică a algoritmului împărțirii întregi este prezentată în figura IV.1.

Pentru a simplifica trasarea legăturilor, precum și pentru a ușura urmărirea succesiunii pașilor unui algoritm se obișnuiește ca aceste linii de legătură să fie parcurse de sus în jos sau de la stînga la dreapta. Liniile de legătură care necesită să fie parcurse în alt mod trebuie să fie desenate sub formă de săgeți. Este însă recomandabil ca toate liniile să fie trasate sub formă de săgeți.

Pentru a nu diminua claritatea unei scheme logice se recomandă evitarea intersectării liniilor de legătură. Dacă totuși unele intersectări nu pot fi eliminate, există un simbol special, denumit conector, care permite întreruperea unei linii de legătură. Un conector este alcătuit dintr-un cerculeț în care se înscrie o literă sau cifră. Se presupune că între două simboluri conector care conțin

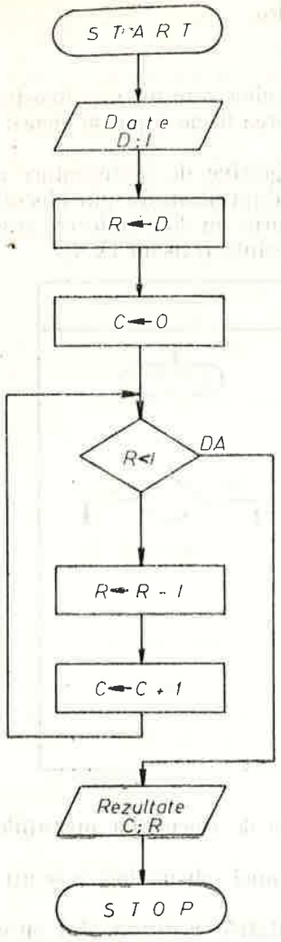


Fig. IV.1

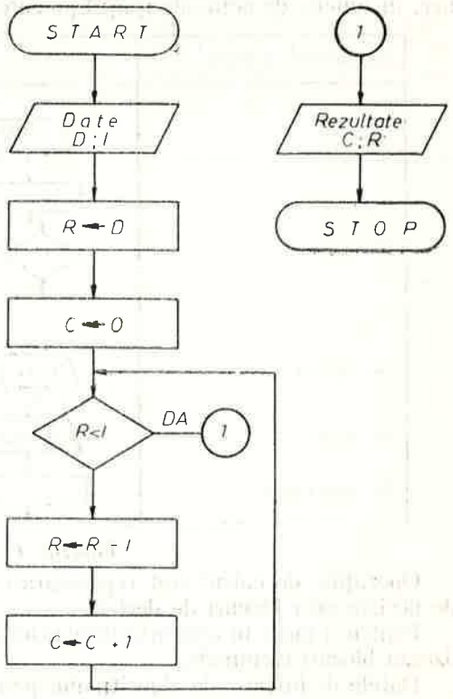


Fig. IV.2

aceeaşi literă sau aceeaşi cifră există o linie de legătură netrasată. În figura IV.2 se prezintă un exemplu de utilizare a acestui tip de simboluri.

Pe măsură ce un algoritm este mai complex, schema sa logică cuprinde mai multe blocuri, iar legăturile necesare sînt mai numeroase. Pentru a simplifica descrierea unui algoritm mai complex, se pot folosi operaţii complexe, a căror efectuare să reprezinte de fapt execuţia unui întreg algoritm. De exemplu, în cadrul algoritmului lui Euclid se poate folosi operaţia de împărţire cu rest, pentru a cărei efectuare se poate utiliza algoritmul împărţirii întregi prezentat anterior. Pentru reprezentarea acestor operaţii speciale se foloseşte în schemele logice un tip de bloc denumit bloc de procedură (vezi tabelul IV.4). În figura IV.3 se poate urmări reprezentarea algoritmului lui Euclid în care s-a utilizat algoritmul împărţirii cu rest. Se observă că blocul de procedură din figura IV.3 descrie aplicarea algoritmului împărţirii întregi pentru valorile variabilelor M şi N, obţinînd ca rezultate valorile variabilelor P şi Q.

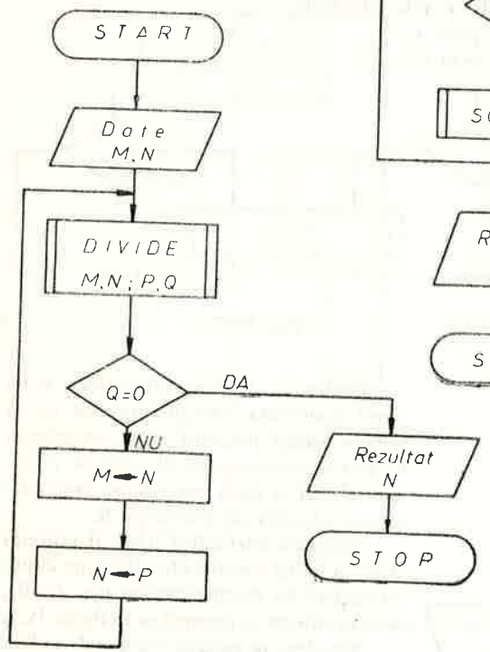


Fig. IV.3

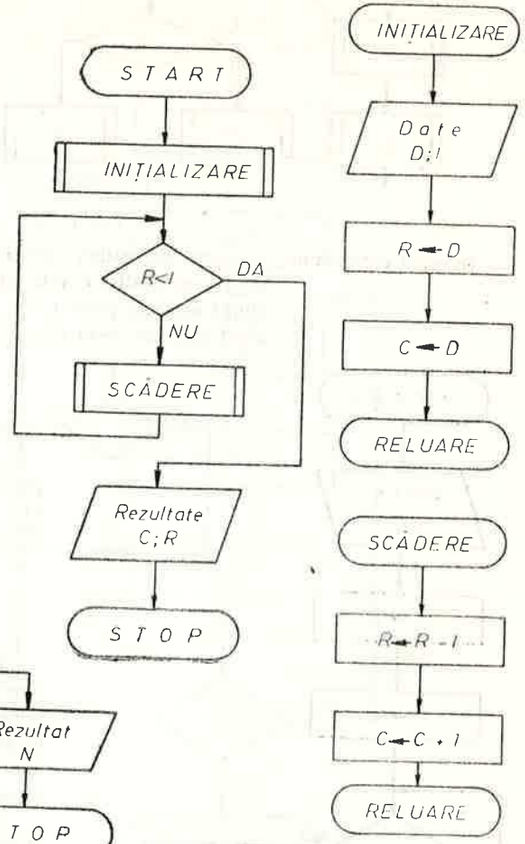


Fig. IV.4

Un bloc de procedură poate fi însă folosit şi pentru a nota o porţiune dintr-o schemă logică, porţiune care este descrisă separat.

Cu o notaţie de acest fel algoritmul împărţirii întregi apare ca în figura IV.4.

Realizarea unui algoritm corect şi a unei scheme logice clare într-un timp scurt şi cu un efort cît mai mic constituie scopul cercetărilor în domeniul programării. S-au obţinut bune rezultate folosind metoda programării structurate, apărută în 1965.

Conform metodei programării structurate un algoritm poate să fie realizat folosind o combinaţie de trei tipuri de structuri elementare: liniară, alternativă şi repetitivă (fig. IV.5). S-a arătat că orice algoritm descris de o schemă logică poate fi transformat astfel ca să fie reprezentat printr-o schemă logică structurată. În figura IV.6 se prezintă algoritmul împărţirii întregi în forma structurată. Singura modificare necesară (faţă de fig. IV.1) a fost la blocul de decizie, unde relaţia respectivă s-a înlocuit cu relaţia complementară.

Structura alternativă corespunde în limbajul curent unei propoziţii condiţionale de forma:

dacă C atunci B altfel A.

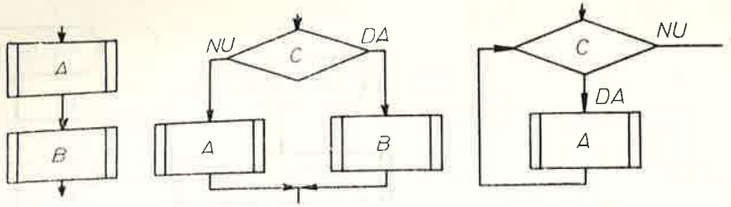


Fig. IV.5

În cazul algoritmilor, structura alternativă corespunde unei operații de decizie de forma

dacă propoziția C este adevărată,
atunci urmează pasul i;
altfel urmează pasul j.

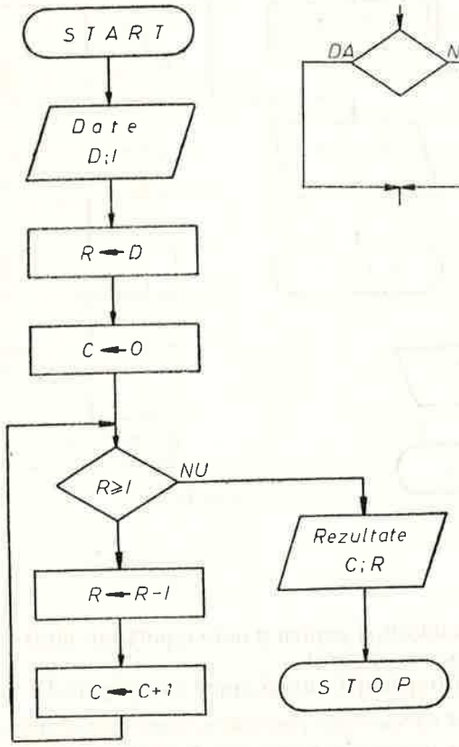


Fig. IV.6

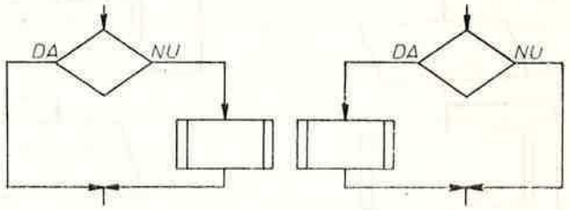


Fig. IV.7

Pentru ca structura alternativă să fie corect construită este însă necesar ca să urmeze același pas, atît după succesiunea de pași corespunzătoare blocului de procedură A, cît și după succesiunea corespunzătoare blocului de procedură B.

Structura alternativă poate fi utilizată și sub o formă simplificată, dacă una dintre alternative nu conține nici un pas. Grafic, această situație se prezintă ca în figura IV.7.

Structura respectivă corespunde în limbajul curent unei propoziții de forma: atît timp cît este adevărată C execută A.

În cazul algoritmilor structura repetitivă corespunde ciclurilor de pași, propoziția C determinînd numărul de repetări ale ciclului.

IV.2.2. Exerciții

1. Să se alcătuiască schema logică a unui algoritm care să calculeze $n!$ cînd se dă n.
2. Să se alcătuiască schema logică a algoritmului pentru rezolvarea ecuației
$$ax^2 + bx + c = 0.$$
3. Ce valoare are S cînd se termină execuția algoritmilor reprezentați de schemele logice din figura IV.8.

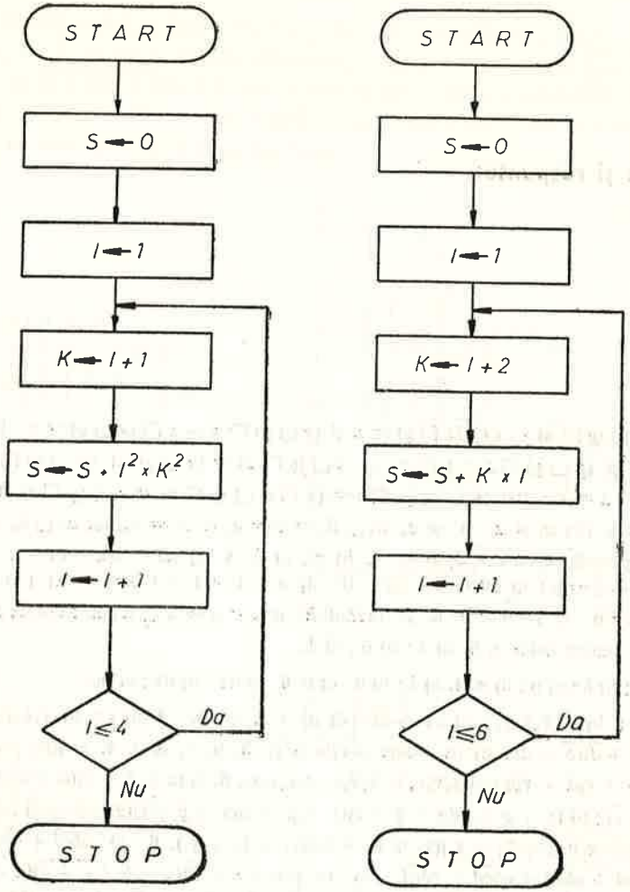


Fig. IV.8

4. Să se alcătuiască schemele logice ale algoritmilor descrișI în exercițiile 4, 5, 6 și 7 din paragraful anterior.

5. Se dă șirul $\{a_k\}$, $k = 1, 2, \ldots$ ai cărui termeni sînt definiți de relația:
$$a_{k+1} = \frac{1}{2}\left(a_k + \frac{b}{a_k}\right);\ a_1 = 1.$$
Să se alcătuiască algoritmul pentru calculul valorii termenului a_n.
Să se deseneze schema logică a acestui algoritm.

6. Să se alcătuiască algoritmii și schemele logice pentru calculul următoarelor sume:
a) $S = 1^2 + 3^2 + 5^2 + \ldots + (2n+1)^2$
b) $S = 1! + 2! + 3! + \ldots + n!$
c) $S = C_n^0 + C_n^1 + C_n^2 + \ldots + C_n^n$.

Indicații și răspunsuri

Cap. I

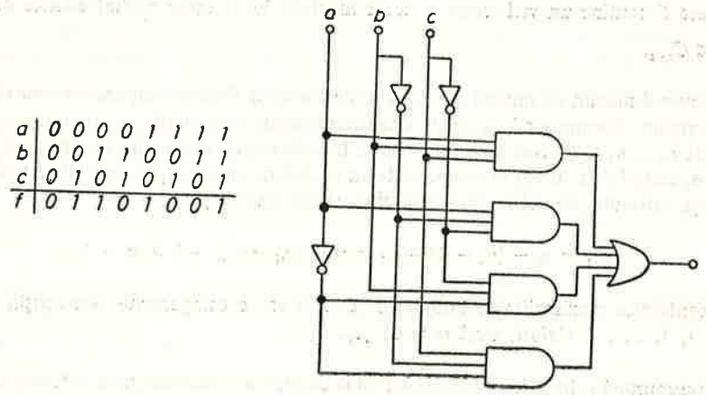
1.1.3. 1. a) $x \sqcup (y \sqcap x) = x \sqcup (x \sqcap y) = x$ și $(x \sqcup y) \sqcap x = x \sqcap (x \sqcup y) = x$; b) $p \sqcup (y \sqcap z) = y \sqcap z$ și $(p \sqcup y) \sqcap z = y \sqcap z$; c) $x \sqcup (y \sqcap u) = (x \sqcup y)$ și $(x \sqcup y) \sqcap u = x \sqcup y$. **2.** $x \sqsubseteq z \Leftrightarrow x \sqcap z = x$ astfel că $(x \sqcup y) \sqcap z = (x \sqcap z) \sqcup (y \sqcap z) = x \sqcup (y \sqcap z)$. **3.** b) $n = 2$; $a_1 = p$ și $a_n = u$. Fie $a_2 \neq a_1$ și $\neq a_n$. $a_2 \sqcup \bar{a}_2 = u = a_n \Rightarrow \bar{a}_2 = a_n$, iar $a_2 \sqcup \bar{a}_2 = p = a_1 \Rightarrow \bar{a}_2 = a_1$ imposibil, deoarece $a_1 \neq a_n$. **4.** b) 8; c) 5. **5.** a) $aa = a(a + 0) = a$; b) $\bar{\bar{a}}a = 0 \Rightarrow \bar{\bar{a}} = a$; c) $a0 = 0 + 0a = 0$; d) $a + 1 = 1 + 1 \cdot a = 1$; e) $1 \cdot 0 = 0$ și $1 + \bar{a} + \bar{a} = 1 \Rightarrow \bar{\bar{a}} = a$; c) $a0 = 0 + 0a = 0$; d) $a + 1 = 1 + 1 \cdot a = 1$; e) $1 \cdot 0 = 0$ și $1 + 0 = 1 \Rightarrow 1 \neq 0$; f) $\bar{1} = \bar{0} = 0$. **6.** $\emptyset$, planul. **8.** $a \cdot n = n \Rightarrow n \sqsubseteq a$ pentru orice a și $a + u = u \Rightarrow a \sqsubseteq u$ pentru orice a. **9.** a) 1; b) 0; c) 1.

1.2.4. 1. a) x; b) b; c) 0; d) x. **4.** a) 1; b) 0; c) 0. **6.** a) da; b) da; c) nu.

1.3.3. 1. a) x; b) $\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$; c) $\bar{a}\bar{b} + \bar{c}\bar{d}$; d) $\bar{a}\bar{c} + \bar{b}\bar{c} + \bar{a}bc$. **2.** a) x; b) $\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$; c) $(\bar{a} + \bar{c})(\bar{b} + \bar{c})(\bar{a} + \bar{d})(\bar{b} + \bar{d})$; d) $(a + b)(b + \bar{c})(\bar{a} + \bar{c})$. **3.** b, c, e, f. **4.** a) $abc + a\bar{b}\bar{c} + \bar{a}\bar{b}\bar{c}$; b) $xyz + \bar{x}yz + \bar{x}\bar{y}\bar{z}$; c) $\bar{x}_1x_2x_3 + x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3$. **5.** a) $(a + b + \bar{c})(a + \bar{b} + c)(a + \bar{b} + \bar{c})(\bar{a} + b + \bar{c})$; b) $(x + y + z)(\bar{x} + y + z)(\bar{x} + y + \bar{z})(x + y + \bar{z})(\bar{x} + \bar{y} + z)$; c) $(x_1 + x_2 + x_3)(\bar{x}_1 + x_2 + x_3)(x_1 + \bar{x}_2 + x_3)(\bar{x}_1 + \bar{x}_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3)$. **6.** a) $\bar{a}\bar{b}cd + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + \bar{a}bcd + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + abc\bar{d}$; b) $(a + b + c + \bar{d})(a + b + \bar{c} + \bar{d})(a + \bar{b} + c + d)(a + \bar{b} + \bar{c} + \bar{d})(\bar{a} + b + c + d)(\bar{a} + b + \bar{c} + d)(\bar{a} + \bar{b} + \bar{c} + d)(\bar{a} + \bar{b} + \bar{c} + \bar{d})$.

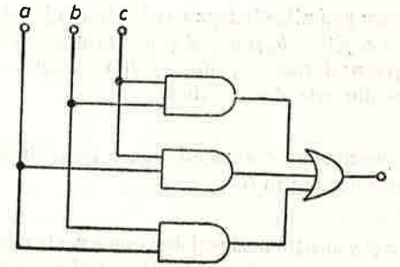
1.4.3. 1. a) 0; b) $x_2\bar{x}_3\bar{x}_4$; c) $x + y + \bar{z}$; d) $\bar{a} + \bar{b} + \bar{c}$. **2.** a) $\bar{a}\bar{b}c + \bar{a}\bar{b}\bar{d}$; b) $xy\bar{z}$; c) $x_1x_2 + \bar{x}_3 + \bar{x}_4$; d) $\bar{a}\bar{b} + \bar{a}cd$; e) $\bar{x}_1$. **3.** a) $\bar{x}\bar{y} + \bar{x}z$; b) $\bar{a}\bar{b} + \bar{a}c + ac$; c) $x_1x_2 + \bar{x}_1x_3$. **4.** a) 1; b) $x_1x_2x_3$; c) $a\bar{b}$. **5.** a) $ac + b\bar{c}$; b) $b\bar{d}$; c) $x + y + \bar{z}$. **6.** b) $\bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$; $(x + y + z)(x + y + \bar{z})(x + \bar{y} + z)(\bar{x} + y + z)$; c) $xy + xz + yz$. **7.** a) $f_1 = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_2x_3 + x_1\bar{x}_2\bar{x}_3 + x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3$, $f_2 = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2x_3 + x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3 + x_1x_2x_3$, $f_3 = \bar{x}_1\bar{x}_2x_3 + x_1\bar{x}_2\bar{x}_3 + x_1x_2\bar{x}_3$; b) $f_1 = (x_1 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2 + x_3)(\bar{x}_1 + \bar{x}_2 + x_3)$, $f_2 = (x_1 + \bar{x}_2 + x_3)(\bar{x}_1 + x_2 + x_3)$, $f_3 = (x_1 + x_2 + x_3)(x_1 + \bar{x}_2 + x_3)(x_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_1 + x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + x_3)$; c) $f_1 = x_2x_3 + x_1x_3 + \bar{x}_2\bar{x}_3$, $f_2 = x_3 + \bar{x}_1\bar{x}_2 + x_1x_2$ $f_3 = x_1\bar{x}_3 + \bar{x}_1\bar{x}_2x_3$. **8.** a) $A\bar{B}\bar{C} + AB\bar{C} + \bar{A}BC + \bar{A}B\bar{C} = A\bar{C} + \bar{A}B$; b) $\bar{A}\bar{B}\bar{C} + \bar{A}BC + ABC = \bar{A}\bar{B}\bar{C} + BC$; c) $AB\bar{C} + A\bar{B}C + \bar{A}BC$. **9.** a) C; b) $\bar{A}\bar{B} + \bar{A}B$; c) $\bar{A}\bar{B}$; d) $A\bar{C} + \bar{A}B + \bar{B}C$. **11.** a) $\bar{a}bcd + a\bar{b}cd + ab\bar{c}d + abc\bar{d} + abcd + abcd + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + ab\bar{c}\bar{d}$; b) $\bar{a}\bar{b}cd + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + ab\bar{c}d + abcd + abc\bar{d} + a\bar{b}\bar{c}d$; c) $\bar{a}\bar{b}\bar{c}de + \bar{a}\bar{b}c\bar{d}e + \bar{a}\bar{b}cd\bar{e} + \bar{a}bc\bar{d}e + \bar{a}bcd\bar{e} + \bar{a}b\bar{c}\bar{d}e + \bar{a}b\bar{c}d\bar{e} + \bar{a}bc\bar{d}\bar{e} + ab\bar{c}\bar{d}e + ab\bar{c}d\bar{e} + abc\bar{d}\bar{e} + a\bar{b}\bar{c}\bar{d}e + a\bar{b}\bar{c}d\bar{e} + a\bar{b}c\bar{d}\bar{e} + ab\bar{c}\bar{d}\bar{e}$. **12.** a) $c + d + a\bar{b}$; b) $\bar{x}\bar{y} + xyz$; c) $\bar{x}z + xw$; d) $x_2\bar{x}_4$; e) $bd + a\bar{b}\bar{c}$.

1.5.6. 1. a) $x_1\bar{x}_2 + x_2x_3 + \bar{x}_1\bar{x}_2 = \bar{x}_2 + x_3$; b) $(x_1\bar{x}_3 + x_2x_3)[\bar{x}_1(\bar{x}_2x_3 + \bar{x}_3)x_4 + x_2\bar{x}_4 + \bar{x}_2x_3] = \bar{x}_2\bar{x}_4(x_1 + x_3)$; c) $(\bar{x}_1 + x_2)(x_1x_2 + \bar{x}_1\bar{x}_3)(x_2\bar{x}_3 + \bar{x}_2x_3) = \bar{x}_1x_2\bar{x}_3$. **2.** a) $\overline{\overline{a_1a_2} \cdot a_3} + a_3a_4 + \overline{a_3 + \overline{a_3 \cdot a_4}}$; b) $\overline{bc + a(b + \bar{c})}(a + b + c) + abc$; c) $[(\bar{x}_3 + \bar{x}_2)x_4 + x_1x_2\bar{x}_4)\overline{x_1\bar{x}_3}$. **3.** a) $\overline{(\bar{a}b) \cdot (b\bar{c})} = a + \bar{b}\bar{c}$; b) $\bar{a}\bar{b}cd + a\bar{b}\bar{c}d + \bar{a}b\bar{c}d + ab\bar{c}\bar{d} = \bar{c}d$; c) $abd + ab\bar{d} + a\bar{b}c + \bar{a}c = ab + c$. **5.** (vezi fig. V.1). **6.** (vezi fig. V.2). **7.** (vezi fig. V.3).



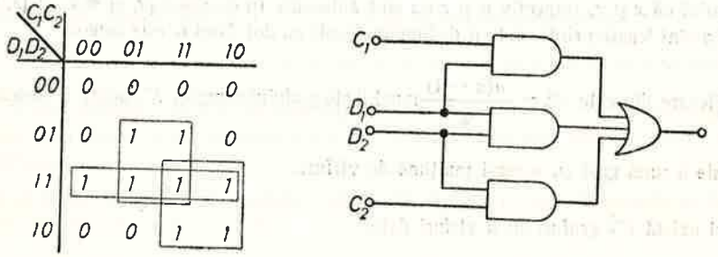
a	0	0	0	0	1	1	1	1
b	0	0	1	1	0	0	1	1
c	0	1	0	1	0	1	0	1
f	0	1	1	0	1	0	0	1

Fig. V.1



a	0	0	0	0	1	1	1	1
b	0	0	1	1	0	0	1	1
c	0	1	0	1	0	1	0	1
f	0	0	0	1	0	1	1	1

Fig. V.2



$f = D_1D_2 + C_1D_1 + C_2D_2$

Fig. V.3

II.1.1. Se obţin 13 lanţuri elementare, 7 care încep cu muchia [1, 2] şi 6 care încep cu muchia [1, 3].

II.1.2. Se va raţiona prin reducere la absurd.

II.1.3. Dacă G conţine un vîrf izolat x, toate muchiile lui G unesc vîrfuri diferite de x, deci $m \leqslant C_{n-1}^2$.

II.1.4. Numărul maxim de muchii ale lui G se obţine cînd fiecare componentă conexă este un subgraf complet. Presupunînd că există două componente care conţin n_1, respectiv n_2 vîrfuri, astfel încît $n_1 \geqslant n_2 \geqslant 2$, vom face următoarea transformare: Suprimăm un vîrf x din componenta cu n_2 vîrfuri şi îl ducem în componenta cu n_1 vîrfuri, unindu-l prin muchii cu toate cele n_1 vîrfuri deja existente. Se obţine un nou graf G_1 cu m_1 muchii şi

$$m_1 = m - (n_2 - 1) + n_1 = m + (n_1 - n_2) + 1 \geqslant m + 1,$$

ceea ce contrazice maximalitatea grafului G. Deci G are p componente care conţin respectiv $n - p + 1$, 1, ..., 1 vîrfuri, cînd $m = C_{n-p+1}^2$.

II.1.5. Presupunînd prin reducere la absurd că G conţine n vîrfuri de grade diferite, rezultă că şirul gradelor lui G este: 0, 1, 2, ..., $n - 1$. Deci G conţine în acelaşi timp un vîrf izolat şi un vîrf adiacent cu toate celelalte vîrfuri, ceea ce este contradictoriu.

II.1.6. Pentru grafurile din figura II.14 bijecţia f se defineşte de exemplu prin: $f(1) = a$, $f(2) = c$, $f(3) = e$, $f(4) = b$, $f(5) = d$ şi pentru grafurile din figura II.15 prin: $f(1) = a$, $f(2) = b$, $f(3) = c$, $f(4) = d$, $f(5) = e$, $f(6) = f$, $f(7) = h$, $f(8) = j$, $f(9) = g$, $f(10) = i$, urmărind ciclurile elementare din cele două grafuri.

II.1.7. Să presupunem că G nu este conex. Dacă vîrfurile x şi y din $\bar{G}$ nu sînt adiacente, rezultă că ele sînt adiacente în G.

Deci x şi y aparţin unei aceleiaşi componente conexe a lui G. Graful G mai conţine cel puţin o altă componentă conexă, deci există un vîrf z care nu aparţine în graful G unei aceleiaşi componente cu x şi cu y.

Rezultă că x şi z, respectiv y şi z nu sînt adiacente în G, adică $[x, z] \in \bar{U}$ şi $[z, y] \in \bar{U}$, deci x şi y sînt legate printr-un lanţ de lungime egală cu doi. Deci $\bar{G}$ este conex.

II.1.8. Fiecare din cele $C_n^2 = \dfrac{n(n-1)}{2}$ muchii ale grafului complet K_n poate fi aleasă sau nu ca muchie a unui graf cu aceeaşi mulţime de vîrfuri.

Deci există $2^{C_n^2}$ grafuri cu n vîrfuri date.

II.1.9. Din definiţia dată rezultă că relaţia binară este simetrică şi reflexivă. Dacă $L_1 = [x, \ldots, y]$ şi $L_2 = [y, \ldots, z]$ rezultă că $L = [x, \ldots, y, \ldots, z]$ este un lanţ de la x la z, deci relaţia binară $\sim$ este tranzitivă.

II.1.10. Să presupunem, prin reducere la absurd, că orice vîrf x al lui G are gradul $d(x) \geqslant 3$.

Fie:

$$L = [x_1, \ldots, x_p]$$

unde $p \geqslant 4$, un cel mai lung lanţ elementar care pleacă din x_1. Rezultă că x_p este adiacent numai cu vîrfuri din mulţimea $\{x_1, \ldots, x_{p-1}\}$. Vîrful x_p are gradul cel puţin 3 şi este adiacent cu x_{p-1}, deci există doi indici $1 \leqslant r < s \leqslant p - 2$ astfel încît x_p să fie adiacent cu x_r şi x_s. Ciclurile elementare $C_1 = [x_s, x_{s+1}, \ldots, x_p, x_s]$ şi $C_2 = [x_r, x_{r+1}, \ldots, x_s, x_p, x_r]$ sînt impare conform ipotezei. Însă ciclul elementar $C_3 = [x_r, x_{r+1}, \ldots, x_{p-1}, x_p, x_r]$ este par, ceea ce contrazice ipoteza.

II.1.11. Vîrfurile 4, 5 şi 6 nu pot aparţine două cîte două unui aceluiaşi subgraf complet.

Fiecare din ele aparţine unui număr de 6 subgrafuri complete cu 3 vîrfuri. Se obţin în total 18 subgrafuri complete cu 3 vîrfuri.

II.2.1. Cele două grafuri izomorfe din figura II.15 nu sînt hamiltoniene.

Graful din figura II.21 este hamiltonian, un ciclu hamiltonian fiind:

[1, 2, 12, 11, 10, 9, 3, 4, 5, 6, 7, 8, 1].

II.2.2. Din cele 8 cicluri hamiltoniene ale grafului dat există două cu un cost minim, egal cu 12 unităţi şi anume: $[A, E, B, C, D, A]$ şi $[A, D, C, B, E, A]$. Răspunde cerinţelor problemei ciclul $[A, E, B, C, D, A]$.

II.2.3. Vîrfurile 2 şi 6 avînd gradul egal cu 2, rezultă că ciclul hamiltonian conţine muchiile $[P, 2]$, $[2, 4]$ şi $[5, 6]$, $[6, 7]$.

Se obţin deci patru cicluri hamiltoniene, dintre care ciclul $[P, 2, 4, 5, 6, 7, 8, 11, 9, 10, 1, 3, P]$ are o lungime totală minimă, egală cu 33 unităţi.

II.2.4. Notînd cu $H(n)$ numărul ciclurilor hamiltoniene ale grafului K_n obţinem $H(3) = 1$ şi $H(n + 1) = nH(n)$, deoarece vîrful x_{n+1} poate fi intercalat în n noduri distincte printre vîrfurile $x_1, \ldots, x_n$ ale unui ciclu hamiltonian cu n vîrfuri pentru a obţine un ciclu hamiltonian cu $n + 1$ vîrfuri. Demonstraţia rezultă prin inducţie după n.

II.2.5. Dacă notăm cu X mulţimea vîrfurilor lui K_n, fie C un ciclu elementar cu k vîrfuri ($3 \leqslant k \leqslant n$), care formează mulţimea A, al lui K_n. Considerînd subgraful lui K_n indus de A, el este un graf complet care admite pe C drept ciclu hamiltonian. Deoarece A poate fi aleasă în X în C_n^k moduri, ţinînd seama de problema precedentă, rezultă că numărul ciclurilor elementare cu k vîrfuri ale lui K_n este egal cu

$$C_n^k \frac{(k-1)!}{2} = \frac{1}{2} \cdot \frac{n(n-1)\ldots(n-k+1)}{k}.$$

Însumînd aceste numere pentru $k = 3, \ldots, n$ se obţine rezultatul căutat.

II.3.1. [1, 2, 3, 4, 5, 6, 4, 7, 6, 8, 7, 2, 10, 8, 9, 10, 1]. Acest ciclu nu este unic.

II.3.2. Graful conţine 6 vîrfuri de grad impar şi anume: 2, 3, 4, 7, 8, 9, deci sînt necesare cel puţin 3 noi muchii pentru ca gradele tuturor vîrfurilor să fie pare, deci ca graful obţinut să fie eulerian.

Dacă adăugăm de exemplu muchiile: [2, 9], [3, 7] şi [4, 8], această condiţie va fi verificată şi graful obţinut are un ciclu eulerian.

II.3.3. Se vor adăuga trei noi muchii, de exemplu [1, 2], [6, 10] şi [3, 7].

II.3.4. Dacă G are gradele tuturor vîrfurilor numere pare, rezultă din teorema demonstrată că fiecare componentă conexă diferită de un vîrf izolat conţine un ciclu eulerian. Dar fiecare ciclu

eulerian se poate scrie ca o reuniune de cicluri elementare, parcurgînd ciclul eulerian plecînd dintr-un vîrf x. Pentru a demonstra necesitatea, să observăm că fiecare ciclu elementar care trece printr-un vîrf x al grafului G utilizează două muchii incidente cu x. Considerînd una din muchiile rămase, ea aparţine unui alt ciclu împreună cu o altă muchie incidentă cu x ş.a.m.d. Deci $d(x)$ este par.

II.3.5. Se consideră graful obţinut din G în modul următor : se adaugă un nou vîrf u împreună cu muchiile $[u, x]$ şi $[u, y]$.

II.4.1. Se obţin trei arbori parţiali minimi. Unul dintre ei are muchiile [6, 7], [1, 6], [1, 4], [1, 2], [2, 3] şi [4, 5].

II.4.2. a) $\Rightarrow$ b) rezultă din propoziţia II.4.2.

Dacă are loc b), G fiind conex conţine un arbore parţial A. Deoarece A are $n-1$ muchii rezultă că $G = A$, deci G nu conţine cicluri, adică are loc c). Dacă are loc c), să presupunem prin reducere la absurd că G nu este conex. Rezultă că G are $p \geqslant 2$ componente conexe care conţin respectiv $n_1, \ldots n_p$ vîrfuri. G fiind fără cicluri, fiecare componentă conexă este un arbore, deci G are $(n_1 - 1) + \ldots + (n_p - 1) = n - p < n - 1$ muchii, contradicţie. Deci G este arbore şi c) $\Rightarrow$ a).

II.4.3. G fiind conex conţine un arbore parţial A. Suprimînd cîte un vîrf terminal al lui A şi al subarborilor care se obţin în acest mod găsim un subarbore A_1 al lui A cu k vîrfuri. Subgraful lui G care are aceeaşi mulţime de vîrfuri cu A_1 satisface condiţiile problemei.

II.4.4. Să presupunem prin reducere la absurd că există un graf G cu n vîrfuri şi cel puţin n muchii care nu conţine nici un ciclu.

Rezultă că G are $p \geqslant 1$ componente conexe, fiecare este un arbore şi ele conţin respectiv $n_1, \ldots, n_p$ vîrfuri. Deci G are $(n_1 - 1) + \ldots + (n_p - 1) = n - p \leqslant n - 1$ muchii, contradicţie.

II.4.5. Să presupunem că prin eliminarea anumitor muchii din G am obţinut un graf fără cicluri, cu p componente conexe care sînt arbori şi conţin $n_1, \ldots, n_p$ vîrfuri. Acest graf are $(n_1 - 1) + \ldots + (n_p - 1) = n - p \leqslant n - 1$ muchii. Deci din graful G trebuie eliminate cel puţin $m - (n - 1) = m - n + 1$ muchii, acest număr fiind suficient pentru a obţine un graf fără cicluri dacă se suprimă toate muchiile care nu aparţin unui arbore parţial al lui G.

II.4.6. Condiţia este necesară, deoarece $d_1 + \ldots + d_n = 2m = 2(n-1)$. Suficienţa se demonstrează prin inducţie după n. Pentru $n = 2$ obţinem un singur arbore cu $d_1 = d_2 = 1$. Presupunînd proprietatea adevărată pentru n, fie numerele $d_1 \geqslant \ldots \geqslant d_{n+1} \geqslant 1$ cu $d_1 + \ldots + d_{n+1} = 2n$. Rezultă $d_{n+1} = 1$. Deci $d_1 + \ldots + d_n = 2n - 1$. Obţinem $d_1 \geqslant 2$, deoarece în caz contrar am avea $d_1 + \ldots + d_n = n < 2n - 1$. Notînd $d_1' = d_1 - 1$, rezultă $d_1' + d_2 + \ldots + d_n = 2n - 2$ şi conform ipotezei de inducţie există un arbore A_1 cu gradele celor n vîrfuri egale respectiv cu $d_1 - 1, d_2, \ldots, d_n$.

Arborele căutat A se obţine din A_1 considerînd un nou vîrf pe care îl legăm printr-o muchie de vîrful lui A_1 de grad $d_1 - 1$.

II.5.1. Dacă facem convenţia ca operaţiile de aceeaşi prioritate să se efectueze de la stînga la dreapta, se obţin rezultatele: a) $** + xy + yz + zx$.

b) $+ - * 3 * \uparrow x 3 y * 5 * \uparrow x 2 \uparrow y 2 * 2 * x \uparrow y 3$.

c) $+ + / + xy + yz / + yz + zx / + zx + xy$.

d) $- + + \uparrow a 3 \uparrow b 3 \uparrow c 3 * 3 * a * b c$.

e) $* - * 2 \uparrow x + * 3 y z a - * 14 t 3$.

f) $+ \uparrow a \uparrow bc / x + 2 / yz$.

II.5.2. Se utilizează inducţia după $n \geqslant 2$. Orice expresie aritmetică cu n operanzi conţine $n - 1$ operatori.

II.5.3. a. $5xyz - 4x^2y - 3z$;

b. $\dfrac{x}{y + \dfrac{x^2}{z}}$; c. $x^{a+2b+3c^2}$;

d. $((x + 3a)(y - 1) + 2(x^2 + 5)$.

II.5.4. Deoarece $2^3 < 10 < 2^4$ obţinem un arbore binar complet cu 6 vîrfuri terminale pe nivelul 3 şi 4 vîrfuri terminale pe nivelul 4.

II.5.5. Dacă alfabetul conţine literele $a_1, a_2, \ldots, a_{24}$ în această ordine, deci putem scrie : $a_1 < a_2 < \ldots < a_{24}$, ordinea cuvintelor formate cu aceste litere într-un dicţionar se defineşte astfel : cuvîntul $x = x_1 \ldots x_n$ spunem că este mai mic decît cuvîntul $y = y_1 \ldots y_m$ şi el va fi scris în dicţionar înaintea cuvîntului y dacă :

a) x este un început propriu al lui y, adică $m > n$ şi $x_1 = y_1, \ldots, x_n = y_n$ sau :

b) există un indice $p \geqslant 1$ astfel încît : $x_1 = y_1, \ldots, x_{p-1} = y_{p-1}$ şi $x_p < y_p$.

Această relaţie de ordine este o *relaţie de ordine totală*, adică pentru $x \neq y$ avem sau $x < y$ sau $y < x$.

II.5.6. Întîi se sortează cele 1 000 000 de numere, adică se obţine un şir :

$$a_1 \leqslant a_2 \leqslant \ldots \leqslant a_{1\,000\,000}.$$

Apoi se parcurge secvenţial lista acestor numere, mărind cu o unitate valoarea unui contor c de fiecare dată cînd întîlnim un nou număr, conform următorului algoritm :

1. $c \leftarrow 1$;
2. $i \leftarrow 1$;
3. $i = 1\,000\,000$? Dacă da, ne oprim. Dacă nu, trecem la pasul următor ;
4. $a_{i+1} > a_i$? Dacă da, $c \leftarrow c + 1$ şi mergem la 5. Dacă nu, mergem la 5.
5. $i \leftarrow i + 1$ şi se merge la 3.

La sfîrşitul aplicării algoritmului valoarea variabilei c va indica răspunsul la problemă.

II.5.7. Dacă comparăm toate cele C_{4001}^2 perechi de cuvinte binare trebuie să facem 8 002 000 comparaţii.

Pentru a reduce numărul comparaţiilor, procedăm astfel : Se sortează întîi fişierul, astfel încît $a_1 < a_2 < \ldots < a_{4001}$. Trebuie să verificăm condiţia $a_i + a_j = c$, unde suma se face în baza 2 şi $c = 11 \ldots 1_2$. Ţinînd seama că fişierul este sortat, putem proceda astfel :

1. $i \leftarrow 1$, $j \leftarrow 4\,001$ şi apoi se repetă pasul 2 pînă cînd $j \leqslant i$;
2. Dacă $a_i + a_j = c$, se tipăreşte perechea $\{a_i, a_j\}$. Se stabileşte $i \leftarrow i + 1$, $j \leftarrow j - 1$;
   Dacă $a_i + a_j < c$, se face $i \leftarrow i + 1$;
   Dacă $a_i + a_j > c$, se face $j \leftarrow j - 1$.

Se observă că la pasul 2 se fac cel mult $n - 1$ comparaţii cu c, dacă sînt n cuvinte în fişier. Deci în cazul nostru se fac cel mult 4 000 comparaţii. Pentru a sorta cele 4 001 cuvinte binare, considerate ca numere scrise în baza 2, mai sînt necesare cel mult 48 000 comparaţii. Rezultă un total de cel mult 52 000 comparaţii.

II.5.8. Se construieşte un arbore binar complet de sortare cu vîrfurile terminale pe cel mult două nivele consecutive. Cei n jucători se asociază celor n vîrfuri terminale, fiecare nod tată avînd asociat cîştigătorul meciului direct dintre cei doi fii ai săi. Cîştigătorul turneului va fi asociat rădăcinii arborelui. Pentru a găsi al II-lea cel mai bun jucător trebuie să joace între ei

învinşii campionului. Am văzut că sînt suficiente $r = [\log_2 n]$ comparaţii la nivelele $r, r-1, \ldots$ $\ldots, 1$, deci $[\log_2 n]$ meciuri. Dacă $n = 2^r$, acest număr de meciuri se reduce la $r - 1$, deoarece toate nodurile terminale sînt la nivelul r, deci trebuie făcute noi comparaţii la nivelele $r-1$, $r-2, \ldots, 1$.

II.5.9. Algoritmul schimbă între ele oricare două numere vecine, dintre care cel mai mare se găseşte înaintea celui mai mic. La fiecare etapă indicele i are proprietatea că înregistrările $a_{i+1}, \ldots, a_n$ se găsesc în poziţia lor finală. Dacă $t = 0$, înseamnă că şirul de numere obţinut verifică proprietatea $a_j \leqslant a_{j+1}$ pentru $j = 1, \ldots, i-1$. Cum numerele $a_{i+1}, \ldots, a_n$ se găsesc în poziţiile finale, rezultă că şirul este sortat şi ne oprim.

Dacă definim o *inversiune* ca fiind o pereche de numere $\{a_i, a_j\}$ cu $i < j$ şi $a_i > a_j$, rezultă că şirul $a_1, \ldots, a_n$ care trebuie sortat prezintă cel mult C_n^2 inversiuni, în cazul cînd $a_1 > a_2 > \ldots > a_n$. Şirul $a_1, \ldots, a_n$ este sortat dacă şi numai dacă el prezintă zero inversiuni.

Dacă $a_j > a_{j+1}$, prin interschimbarea numerelor a_j şi a_{j+1} între ele, şirul obţinut are cu o inversiune mai puţin decît şirul de la care am plecat. Deci după un număr finit de interschimbări obţinem un şir cu zero inversiuni, deci un şir sortat.

II.5.10. Se compară primele numere din cele două şiruri sortate, iar cel mai mic se trece în şirul final. Se repetă operaţia pentru şirurile obţinute ş.a.m.d. Cînd unul din şiruri este epuizat, toate numerele rămase în celălalt se trec în şirul final, ele fiind cele mai mari elemente din cele două şiruri. Putem deci scrie:

1. Se stabileşte $i \leftarrow 1$, $j \leftarrow 1$, $k \leftarrow 1$.
2. Dacă $a_i \leqslant b_j$, mergi la pasul 3, altfel treci la pasul 5.
3. Se fac atribuirile $c_k \leftarrow a_i$, $k \leftarrow k+1$, $i \leftarrow i+1$. Dacă $i \leqslant m$, mergi la 2. În caz contrar, treci la pasul următor.
4. Stabileşte $(c_k, \ldots, c_{m+n}) \leftarrow (b_j, \ldots, b_n)$ şi termină algoritmul.
5. Stabileşte $c_k \leftarrow b_j$, $k \leftarrow k+1$, $j \leftarrow j+1$. Dacă $j \leqslant n$, mergi la 2, altfel treci la pasul următor.
6. Stabileşte $(c_k, \ldots, c_{m+n}) \leftarrow (a_i, \ldots, a_m)$ şi termină aplicarea algoritmului.

La paşii 4, respectiv 6 ajungem cînd unul din şiruri este parcurs în întregime, copiind în şirul final, ceea ce a mai rămas din celălalt şir.

La fiecare comparaţie de la pasul 2 transmitem în şirul final $c_1, \ldots, c_{m+n}$ cel puţin un număr, la paşii 3 sau 5. Deoarece ultimul număr este trecut în şirul final fără nici o comparaţie, rezultă că acest algoritm de interclasare necesită cel mult $m + n - 1$ comparaţii.

Acest număr maxim de comparaţii este atins cînd de exemplu avem $\max(a_1, \ldots, a_{m-1}) \leqslant b_1$ şi $a_n > b_n$.

II.5.11. Algoritmul de căutare este următorul:

1. $i \leftarrow 1$.
2. $b = a_i$? Da: Stop. Căutarea a avut succes. Nu: Treci la pasul următor.
3. $b < a_i$? Da: Stop. Căutarea nu a avut succes. Nu: Treci la pasul următor.
4. $i = 4$? Da: Stop. Căutarea nu a avut succes şi $b > a_4$.

Nu: $i \leftarrow i + 1$ şi se merge la pasul 2.

II.5.12. În cazul arborelui de căutare din figura II.48 numărul total de comparaţii în cazul unei căutări cu succes este egal cu 8.

Deci numărul mediu de comparaţii în cazul unei căutări cu succes este egal cu $\frac{8}{4} = 2$ comparaţii.

Dacă $b < a_1$ ajungem în nodul terminal numerotat cu 0 şi facem 2 comparaţii: cu a_2 şi cu a_1 etc.

Deci numărul total de comparaţii pentru căutările fără succes este egal cu 12, iar numărul mediu este egal cu $\frac{12}{5} = 2{,}4$ comparaţii.

Pentru arborele de căutare din figura II.49 obţinem în mod analog numărul mediu de comparaţii pentru căutarea cu succes egal cu 2,5 comparaţii. Pentru căutarea fără succes numărul mediu de comparaţii este egal cu 2,8.

Rezultă că arborele de căutare din figura II.48 este superior arborelui din figura II.49 în ceea ce priveşte realizarea unui număr mediu cît mai mic de comparaţii.

II.5.13. Se poate aplica următoarea idee: Ori de cîte ori o înregistrare a fost localizată cu succes, ea este deplasată la începutul benzii, provocînd translaţia celorlalte înregistrări, în aceeaşi ordine, către sfîrşitul benzii.

După ce s-au făcut un număr de căutări cu succes ale înregistrărilor de pe bandă, înregistrările mai des căutate vor ocupa poziţii mai apropiate de începutul benzii.

Cap. III

III.1.1. Se găsesc 4 drumuri care folosesc arcul (1, 3) şi 4 drumuri care folosesc arcul (1, 2).

Singurele circuite elementare sînt circuitele de lungime 2: (3, 4, 3) şi (5, 6, 5).

III.1.2. Atît în suma gradelor de intrare cît şi în suma gradelor de ieşire fiecare arc este numărat o dată şi numai o dată, deoarece iese dintr-un vîrf şi intră într-un singur vîrf.

III.1.3. Oricare pereche de vîrfuri $\{x, y\}$ cu $x \neq y$ poate fi unită prin zero arce, prin arcul (x, y), prin arcul (y, x) sau prin ambele arce (x, y) şi (y, x). Rezultă în total $4^{C_n^2} = 2^{n^2-n}$ grafuri orientate cu n vîrfuri. Dacă graful este complet prima posibilitate este exclusă. Deci există $3^{C_n^2}$ grafuri orientate şi complete cu n vîrfuri.

III.1.4. Conform definiţiei, relaţia binară este relativă şi simetrică. Pentru a arăta că este tranzitivă, se observă că dacă D_1 este un drum de la x la y şi D_2 este un drum de la y la z, atunci există un drum D_3 de la x la z obţinut din şirul vîrfurilor lui D_1 care se continuă cu şirul vîrfurilor din D_2: $D_3 = (x, \ldots, y, \ldots, z)$.

În mod analog se arată că există un drum şi de la z la x.

Pentru graful din figura III.1 componentele tare conexe sînt următoarele: {1, 2, 3}, {4}, {5}, {6, 7, 8}.

III.1.5. Dacă D nu este drum elementar, fie z primul vîrf întîlnit în sensul de la x la y care se repetă în şirul de vîrfuri care îl defineşte pe D. Vom suprima din D subşirul de vîrfuri cuprins între prima şi ultima apariţie a lui z în şirul D, păstrînd în D o singură apariţie a lui z. Aplicăm acelaşi procedeu pentru drumul obţinut, pînă la găsirea unui drum elementar de aceleaşi extremităţi.

III.1.6. Vom presupune prin reducere la absurd că graful-turneu G nu conţine un drum elementar care trece prin toate vîrfurile grafului.

Fie $D = (x_1, x_2, \ldots, x_p)$ un cel mai lung drum elementar al lui G. Va exista un vîrf y diferit de $x_1, \ldots, x_p$. Graful fiind complet, y este legat prin arce cu vîrfurile $x_1, \ldots, x_p$.

Dacă există arcul (y, x_1) obţinem un drum mai lung ca D, ceea ce contrazice presupunerea făcută.

Deci există arcul (x_1, y). Analog, dacă există arcul (x_p, y), obţinem un drum mai lung ca D. Deci există arcul (y, x_p). Vor exista deci două vîrfuri consecutive x_k şi x_{k+1} ale lui D pentru care arcele care le unesc cu y să aibă sensuri contrare şi anume să existe arcele (x_k, y) şi (y, x_{k+1}) Am obţinut iarăşi un drum mai lung ca D şi anume: $(x_1, x_2, \ldots, x_k, y, x_{k+1}, \ldots, x_p)$, ceea ce este contradictoriu.

Deci D conţine toate vîrfurile grafului G.

III.1.7. Egalitatea 1) exprimă faptul că fiecare vîrf este incident cu exact $n - 1$ arce. Pentru a demonstra 2) se ține seama de problema III.1.2 și de faptul că orice graf-turneu are C_n^2 arce. Pentru a demonstra 3) se înmulțește 1) cu r_i, respectiv cu s_i și se însumează egalitățile obținute, ținînd seama de 2).

III.1.8. Se va alege pentru x un vîrf pentru care $d^+(x)$ este maxim în G.

III.2.1. Se obțin două drumuri critice de lungime egală cu 16 și anume: (1, 2, 5, 8) și (1, 3, 7, 8). Deci data terminării lucrării este $t_8 = 16$.

III.2.2. Se obține graful programului din figura V.4. Drumul critic este (1, 2, 3, 5, 6). Intervalele de fluctuație au fost reprezentate în pătrățele asociate vîrfurilor și marginile totale ale operațiilor au fost trecute în paranteze lîngă arcele respective.

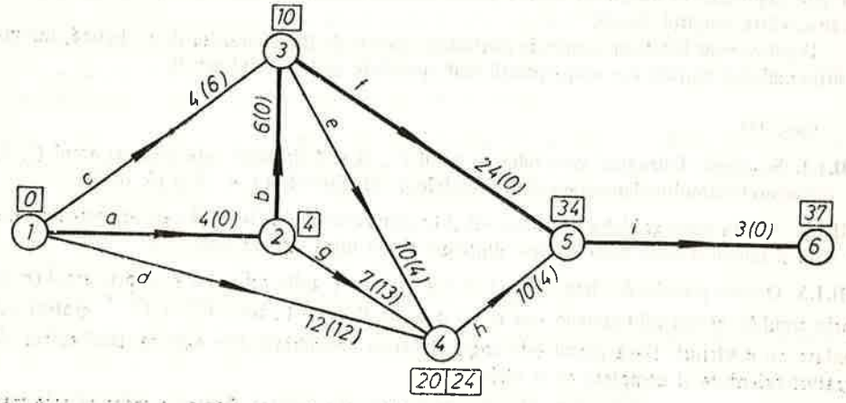


Fig. V.4

III.2.3. Să presupunem prin reducere la absurd că pentru orice vîrf x al grafului avem $p(x) \neq \emptyset$. Fie $D = (x_1, x_2, \ldots, x_p)$ un drum elementar care are un număr maxim de arce în graful G. Rezultă că $p(x_1) \subset \{x_2, \ldots, x_p\}$, deci există un indice k astfel încît $2 \leqslant k \leqslant p$ și $x_k \in p(x_1)$. Am obținut un circuit elementar în graful G:

$$C = (x_1, x_2, \ldots, x_k, x_1),$$

ceea ce contrazice ipoteza că G este fără circuite. În mod analog se demonstrează existența unui vîrf y cu proprietatea $s(y) = \emptyset$, considerînd extremitatea x_p a drumului D.

III.2.4. Deoarece fiecare din cele două arce disjunctive poate avea două orientări, rezultă în total 4 grafuri de activități. Dintre acestea, graful cu o lungime minimă a drumului critic, egală cu 16 unități, se obține dacă alegem arcul (3, 2) cu durata 6 și arcul (5, 4) cu durata 3. Deci trebuie ca evenimentul 3 să preceadă evenimentul 2 și evenimentul 5 să preceadă evenimentul 4.

III.3.1. $\max f_b = 9$.

III.3.2. Se găsește fluxul maxim în rețeaua obținută din rețeaua dată prin înmulțirea capacităților tuturor arcelor cu $[2, 3, 11] = 66$. Apoi se revine la rețeaua inițială, împărțind toate componentele pe arce ale fluxului maxim astfel obținut prin 66. Se găsește $\max f_b = \frac{69}{22}$ și o tăietură de capacitate minimă are suportul $A = \{2, 3, 4, 5, b\}$.

III.3.3. Rețeaua de transport asociată problemei are vîrfurile $x_1, x_2, x_3, y_1, y_2, y_3$, intrarea a și ieșirea b. Arcele (a, x_i) au o capacitate egală cu a_i și arcele (y_i, b) au o capacitate egală cu b_i pentru $i = 1, 2, 3$.

Arcelor de forma (x_i, y_j) le vom asocia o capacitate egală cu cantitatea de produse care pot fi transportate din x_i în y_j. Un flux compatibil cu capacitățile arcelor reprezintă un plan de transport posibil. Fluxul f_b va reprezenta cantitatea totală de produse transportate din centrele x_i în centrele y_j. Aplicînd algoritmul lui Ford-Fulkerson se găsește un flux maxim ale cărui componente pe arcele (x_i, y_j) ne dau următorul plan optim de transport, care utilizează tot disponibilul de la centrele x_1, x_2, x_3:

	y_1	y_2	y_3
x_1	12	2	46
x_2	14	0	10
x_3	7	17	12

Cantitatea de produse transportate este egală cu $f_b = 33 + 19 + 68 = 120$ unități. Soluția problemei nu este unică.

III.3.4. Deoarece $a \notin A$ și $b \in A$, rezultă că drumul D va conține cel puțin o pereche de vîrfuri vecine x_i și x_{i+1} astfel încît $x_i \notin A$ și $x_{i+1} \in A$. În acest caz arcul $(x_i, x_{i+1}) \in \omega^-(A)$.

III.3.5. Graful din figura III.15 este o rețea de transport, unde costurile de instalare a posturilor de control devin capacități ale arcelor.

În această rețea trebuie să determinăm o mulțime de arce astfel încît orice drum de la a la b să conțină cel puțin un arc din această mulțime, iar suma capacităților acestor arce să fie minimă.

Această mulțime de arce formează o tăietură de capacitate minimă.

Aplicînd algoritmul lui Ford-Fulkerson găsim o tăietură de capacitate minimă, de suport $A = \{4, b\}$, deci $\omega^-(A) = \{(1, 4,) (5, 4), (5, b), (6, b)\}$ și $c(\omega^-(A)) = 2 + 2 + 6 + 5 = 15$ unități. Rezultă că posturile de control care supraveghează toate drumurile din a în b trebuie amplasate pe arcele din mulțimea $\omega^-(A)$, cu un cost total de instalare de 15 unități.

III.3.6. Se găsesc activitățile critice ale grafului din figura III.16, care sînt: (1, 2), (2, 3), (3, 6), (3, 4), (4, 5), (5, 6), (5, 7) și (6, 7). Aceste arce formează graful parțial din figura V.5.

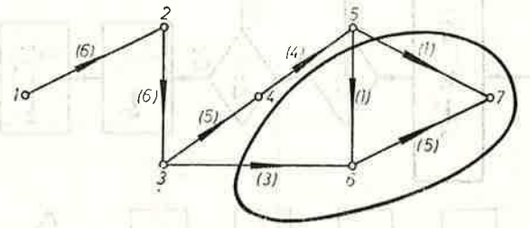


Fig. V.5

Asociind arcelor din figura V.5 costurile din tabelul dat, se obține o rețea de transport cu intrarea 1 și ieșirea 7, pentru care aceste costuri pot fi interpretate drept capacități ale arcelor.

Problema revine la a determina o mulțime de arce, astfel încît orice drum de la 1 la 7 în graful din figura V.5 să conțină cel puțin un arc din această mulțime, iar suma capacităților acestor arce să fie minimă. Conform discuției de la problema precedentă, trebuie să găsim o

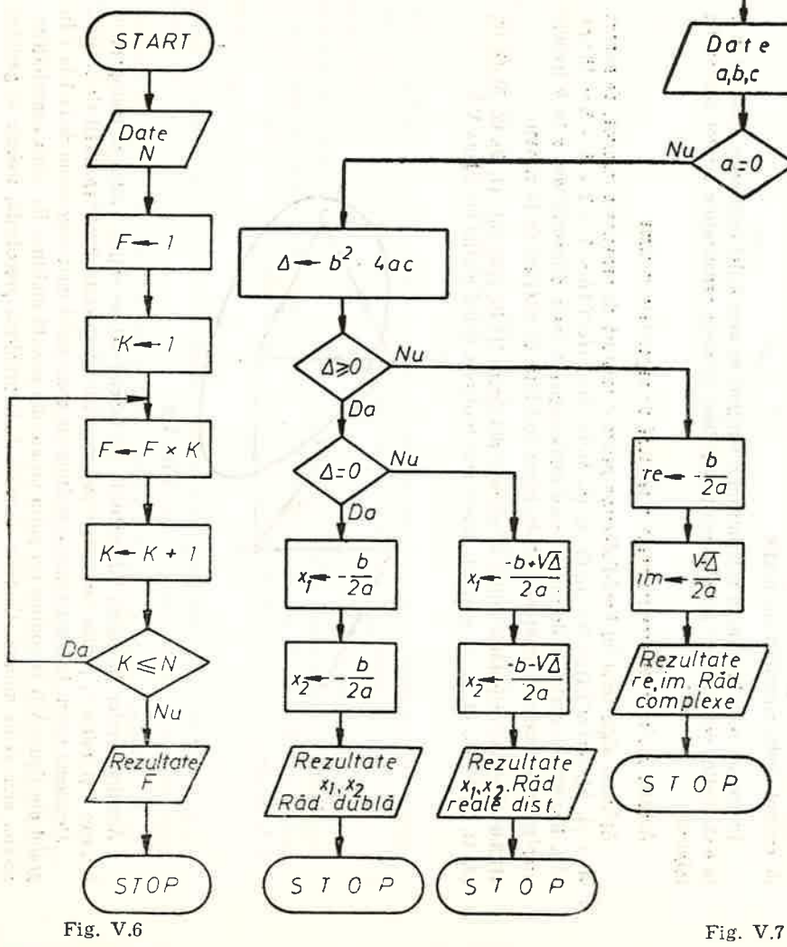


Fig. V.6

Fig. V.7

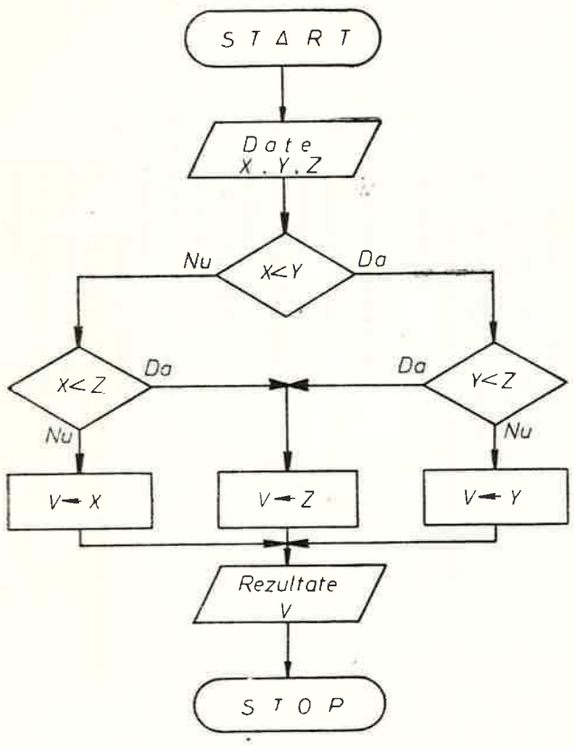


Fig. V.8

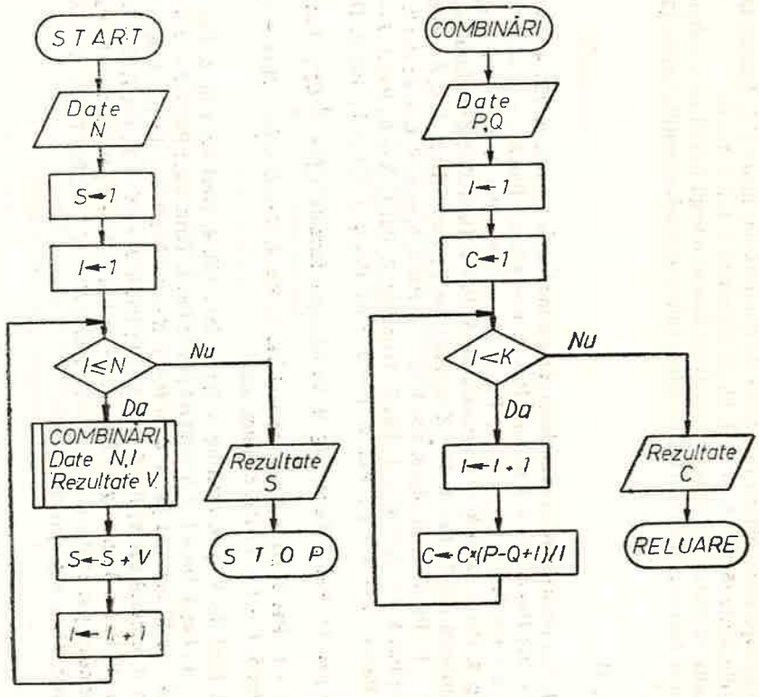


Fig. V.9

tăietură de capacitate minimă în rețeaua de transport din figura V.5. Această tăietură este formată din arcele (5, 7), (5, 6) și (3, 6), de capacitate $1 + 1 + 3 = 5$.

Rezultă că costul minim pentru scurtarea duratei întregii lucrări cu o unitate este egal cu 5 și el se realizează dacă scurtăm cu cîte o unitate duratele operațiilor critice (3, 6), (5, 6) și (5, 7).

Cap. IV

IV.1.5. 1. Nu, deoarece M nu se anulează și algoritmul nu se termină. **2.** 225. **4.** Pas 1. $v \leftarrow a_4$; Pas 2. $p \leftarrow x_0$; Pas 3. $v \leftarrow v + p \times a_3$; Pas. 4. $p \leftarrow p \times x_0$; Pas 5. $v \leftarrow v + p \times a_2$; Pas 6. $p \leftarrow p \times x_0$; Pas 7. $v \leftarrow v + p \times a_1$; Pas 8. $p \leftarrow p \times x_0$; Pas 9. $v \leftarrow v + p \times a_0$; Pas 10. Term. alg. **5.** Pas 1. Dacă $X < Y$ urmează pas 5; Pas 2. Dacă $X < Z$ urmează pas 4; Pas 3. $V \leftarrow X$ urmează pas 7; Pas 4. $V \leftarrow Z$ urmează pas 7; Pas. 5 Dacă $Y < Z$ urmează pas 4; Pas 6. $V \leftarrow Y$; Pas 7. Term. alg. **6.** Pas 1. $a_1 \leftarrow a$; Pas 2. $b_1 \leftarrow a_1 \times x_0 + b$; Pas 3. $c_1 \leftarrow b_1 \times x_0 + c$; Pas 4. $r \leftarrow c_1 \times x_0 + d$; Pas 5. Term. alg. **7.** Pas 1. $S \leftarrow 0$; Pas 2. $I \leftarrow 0$; Pas 3. $T \leftarrow a$; Pas 4. $S \leftarrow S + T$; Pas 5. $T \leftarrow T \times q$; Pas 6. $I \leftarrow I + 1$; Pas 7. Dacă $I \leqslant n$ urmează pas 4; Pas 8. Term. alg. **8.** Se folosește formula $C_n^k = \frac{n}{k} C_{n-1}^{k-1}$; Pas 1. $I \leftarrow 0$; Pas 2. $C \leftarrow 1$; Pas 3. Dacă $I = k$ urmează pas 6; Pas 4. $C \leftarrow C \times (n - I)/(k - I)$ urmează pas 3; Pas 5. $I \leftarrow I + 1$; Pas 6. Term. alg.

IV.2.2. 1. (vezi fig. V.6). **2.** (vezi fig. V.7); **3.** 584; 133. **4.** (vezi fig. V.8). **5.** Pas 1. $V \leftarrow 1$; Pas 2. $I \leftarrow 1$; Pas 3. Dacă $I < N$ urmează pas 5; Pas 4. Term. alg.; Pas 5. $I \leftarrow I + 1$; Pas 6. $V \leftarrow 1/2(V + b/V)$ urmează pas 3. **6.** a) Pas 1. $S \leftarrow 1$; Pas 2. $I \leftarrow 1$; Pas 3. $K \leftarrow 1$; Pas 4. Dacă $I \leqslant N$ urmează pas 8; Pas 5. $I \leftarrow I + 1$; Pas 6. $K \leftarrow K + 2$; Pas 7. $S \leftarrow S + K^2$ urmează pas 4; Pas 8. Term. alg. b) Pas 1. $S \leftarrow 0$; Pas 2. $K \leftarrow 1$; Pas 3. $T \leftarrow 1$; Pas 4. Dacă $K > N$ urmează pas 8; Pas 5. $S \leftarrow S + T$; Pas 6. $K \leftarrow K + 1$; Pas 7. $T \leftarrow T \times K$ urmează pas 4. Pas 8. Term. alg. c) (vezi fig. V.9).

Cuprins

I. **ALGEBRĂ BOOLEANĂ** 3

I.1. Noțiunea de algebră booleană 3
I.1.1. Latice 3
I.1.2. Latice distributive și complementate 6
I.1.3. Exerciții 9
I.2. Funcții booleene 10
I.2.1. Expresii booleene 10
I.2.2. Funcții booleene 12
I.2.3. Reprezentarea funcțiilor booleene 13
I.2.4. Exerciții 16
I.3. Forme canonice ale funcțiilor booleene 18
I.3.1. Forme normale ale funcțiilor booleene 18
I.3.2. Forme canonice ale funcțiilor booleene 19
I.3.3. Exerciții 22
I.4. Simplificarea funcțiilor booleene 23
I.4.1. Simplificarea prin calcul boolean 23
I.4.2. Simplificarea prin diagrame 24
I.4.3. Exerciții 29
I.5. Realizarea fizică a funcțiilor booleene 31
I.5.1. Circuite cu contacte 31
I.5.2. Circuite logice simple 33
I.5.3. Proiectarea circuitelor logice 38
I.5.4. Circuite logice complexe 41
I.5.5. Realizarea practică a circuitelor logice 45
I.5.6. Exerciții 46

II. **GRAFURI NEORIENTATE** 49

II.1. Noțiuni de bază 49
Probleme 54
II.2. Grafuri hamiltoniene 56
Probleme 58
II.3. Grafuri euleriene 59
Probleme 61
II.4. Arbori 62
Probleme 66
II.5. Arbori binari și aplicații 67
II.5.1. Notația fără paranteze a unei expresii aritmetice 68
II.5.2. Arbori de sortare 70
II.5.3. Algoritmul de căutare binară 74
Probleme 78

III. **GRAFURI ORIENTATE** 80

III.1. Noţiuni de bază 80
Probleme 82
III.2. Metoda drumului critic 83
III.2.1. Drumul critic într-un graf de activităţi 83
III.2.2. Evenimente critice şi intervale de fluctuaţie 85
III.2.3. Determinarea drumului critic 88
Probleme 91
III.3. Flux maxim într-o reţea de transport 92
Probleme 99

IV. **ALGORITMI ŞI METODE DE REPREZENTARE** 102

IV.1. Noţiunea de algoritm 102
IV.1.1. Definiţia algoritmului 102
IV.1.2. Proprietăţile algoritmilor 104
IV.1.3. Structura algoritmilor 104
IV.1.4. Clasificarea algoritmilor 107
IV.1.5. Exerciţii 108
IV.2. Metode de reprezentare a algoritmilor 109
IV.2.1. Scheme logice 109
IV.2.2. Exerciţii 112
Indicaţii şi răspunsuri 114

Coli de tipar 8. B.T. 30.IV.84.
Format 16/70×100. Apărut 1984.

I. P. „Oltenia" Craiova
Str. M. Viteazul, nr. 4
Republica Socialistă România
Plan 30210/23/1984.